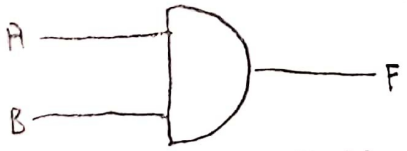
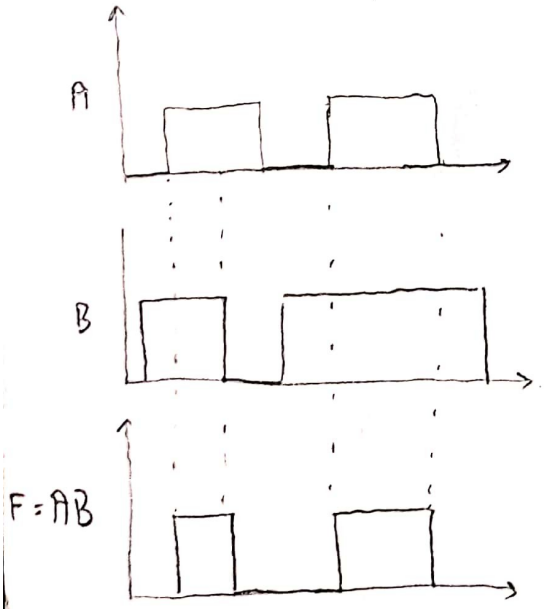
Lojik Notlarım

şafak bilici

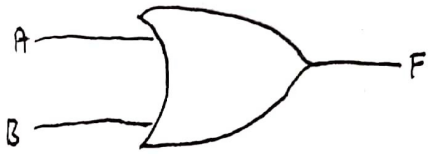
AND



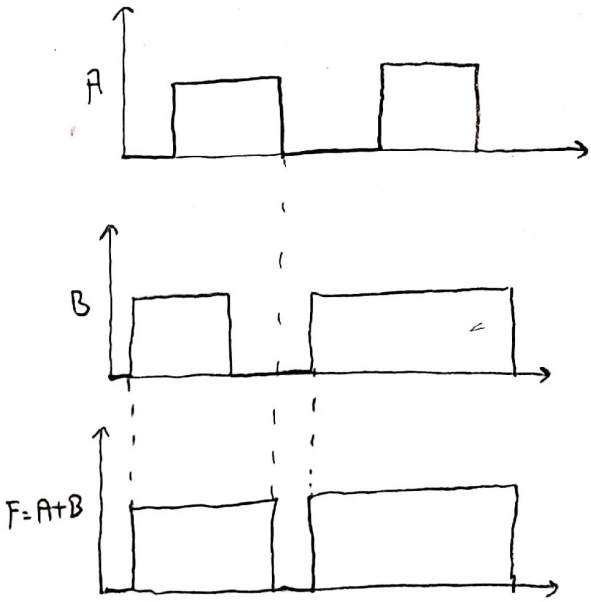
F = AB



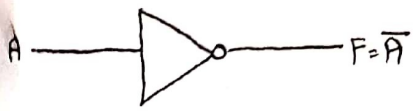
OR

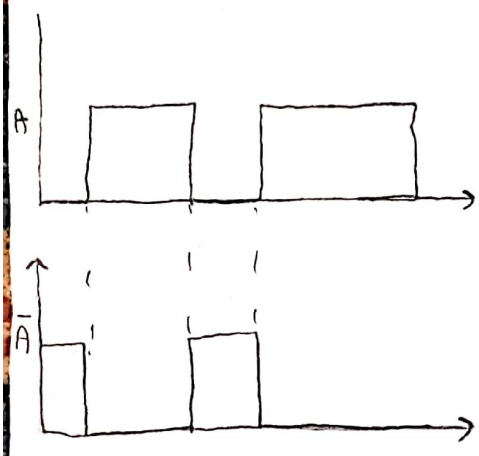


F = A+B

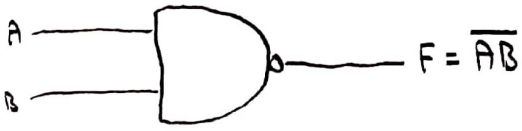


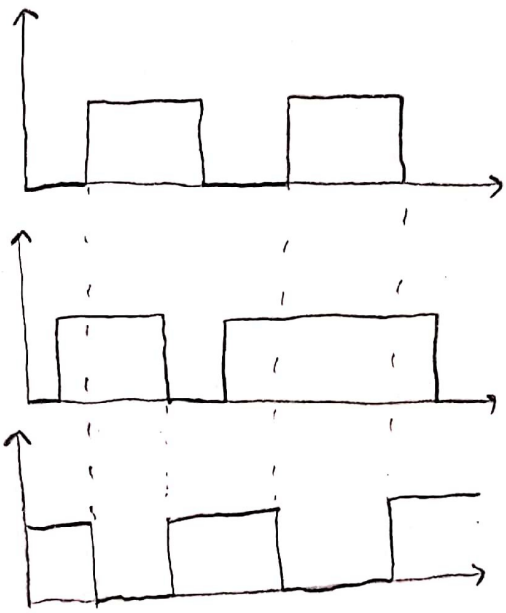
NOT



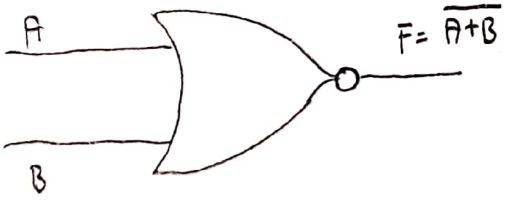


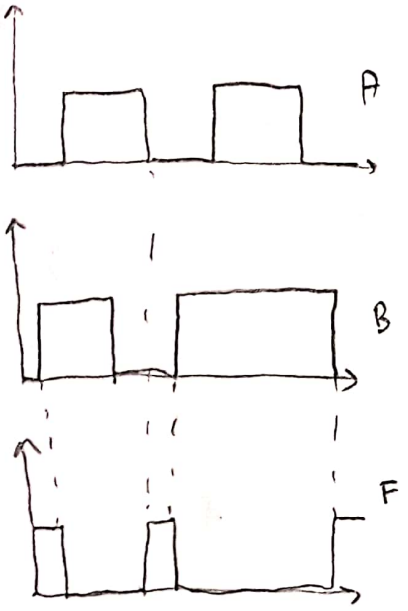
NAND



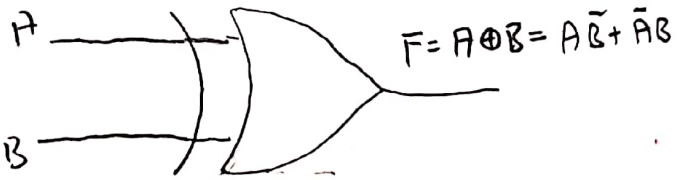


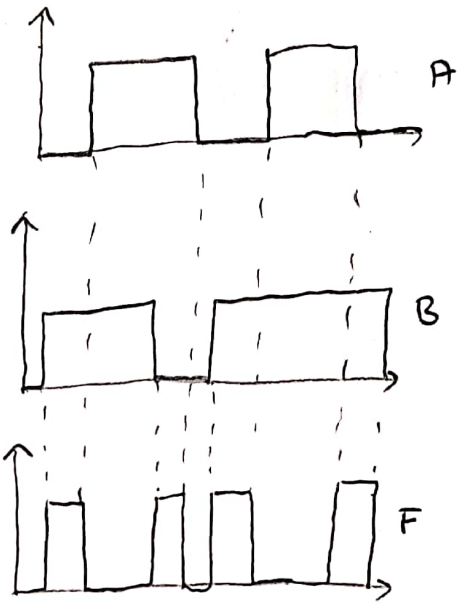
NOR


A
B
F= $\overline{A+B}$

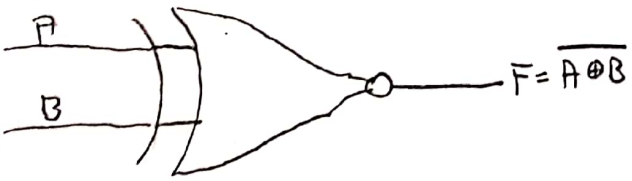

A
B
F

XOR


A
B
F= $A \oplus B = A\bar{B} + \bar{A}B$


A
B
F

XNOR


A
B
F= $\overline{A \oplus B}$

Lojik Fonksiyonların İndirgenmesi

$$f(a,b,c,d) = \bar{b}\,\bar{d} + \bar{b}\cdot\bar{c} + \bar{a}\cdot c\cdot\bar{d}$$

ab \ cd	00	01	11	10
00	1	1	0	1
01	0	0	0	1
11	0	0	0	0
10	1	1	0	1

maxterm cinsinden

$$f(a,b,c,d) = (\bar{c}+\bar{d})\cdot(\bar{a}+\bar{b})\cdot(c+\bar{b})$$

→ Karnoogh

Eksik Boole

$$f(a,b,c,d) = \sum^{1}(5,7,11,12,15) + \sum_{\emptyset}^{!}(0,10,13,14)$$

ab \ cd	00	01	11	10
00	0 Ø	1 0	3 1	2 0
01	4 0	5 0	7 1	6 0
11	12 1	13 Ø	15 1	14 Ø
10	8 0	9 0	11 1	10 Ø

SOR

B=1

14=1

Minterm Ve Maxterm

$\bar{x}\bar{y}$, $\bar{x}y$, $x\bar{y}$, xy ⟶ minterm

$x+y$, $x+\bar{y}$, $\bar{x}+y$, $\bar{x}+\bar{y}$ ⟶ maxterm

x	y	Minterm	m'ye indis	Maxterm	m'ye indis
0	0	$\bar{x}\cdot\bar{y}$	m_0	$x+y$	M_0
0	1	$\bar{x}\cdot y$	m_1	$x+\bar{y}$	M_1
1	0	$x\cdot\bar{y}$	m_2	$\bar{x}+y$	M_2
1	1	$x\cdot y$	m_3	$\bar{x}+\bar{y}$	M_3

Kanonik gösterim:

a	b	F=a+b	Minterm	index
0	0	0	$\bar{a}\cdot\bar{b}$	m_0
0	1	1	$\bar{a}\cdot b$	m_1
1	0	1	$a\cdot\bar{b}$	m_2
1	1	1	$a\cdot b$	m_3

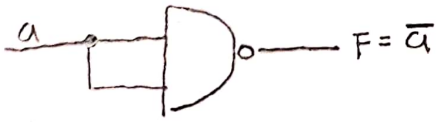
$F=a+b=\bar{a}\cdot b+a\cdot\bar{b}+a\cdot b=m_1+m_2+m_3=\sum(1,2,$

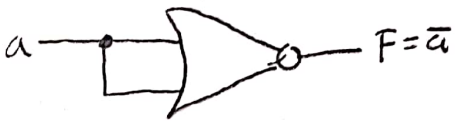
a	b	F=a·b	maxterm	index
0	0	0	$a+b$	M_0
0	1	0	$a+\bar{b}$	M_1
1	0	0	$\bar{a}+b$	M_2
1	1	1	$\bar{a}+\bar{b}$	M_3

$F=a\cdot b=(a+b)\cdot(a+\bar{b})\cdot(\bar{a}+b)=\prod(0,1,2)$

Türetilmiş Kapılarla Temel Lojik Kapıların Yapılması

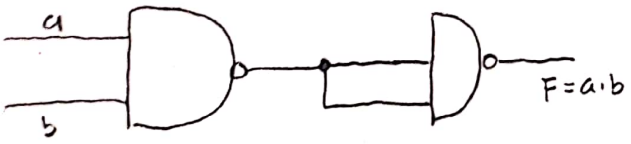
Tümleme



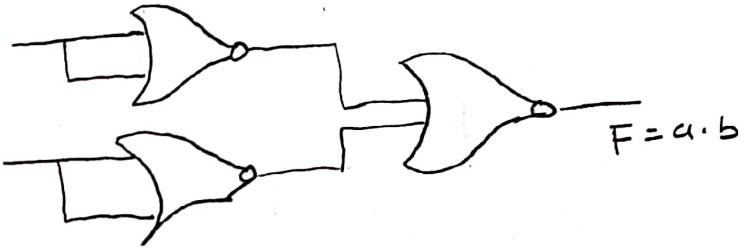


AND

$f = a \cdot b = \overline{\overline{a \cdot b}} = (\overline{\overline{a \cdot b}})$

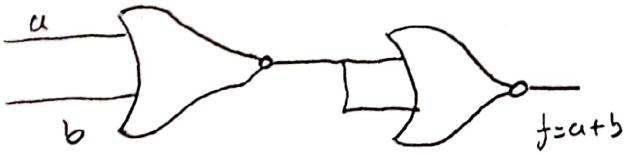


$f = a \cdot b = \overline{(\bar{a} + \bar{b})}$

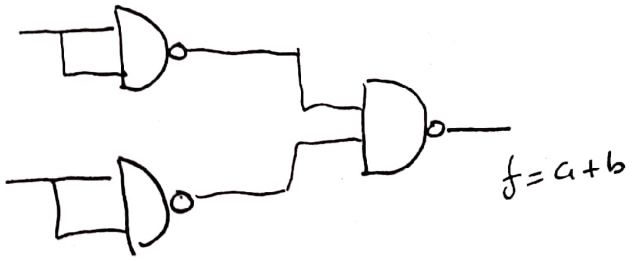


OR

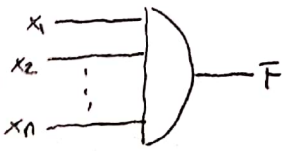
$f = a + b = \overline{\overline{(a + b)}}$

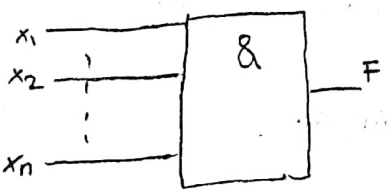


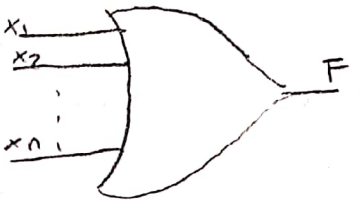
$f = a + b = \overline{(\bar{a} \cdot \bar{b})}$

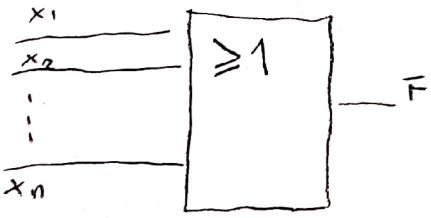


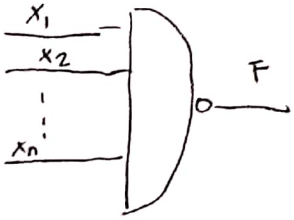
Kombinezonsal Devreler

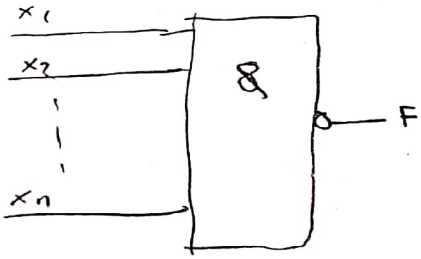
AND ⟶ $F = X_1 \cdot X_2 \cdot X_3 \cdots X_n$ ⟶ 

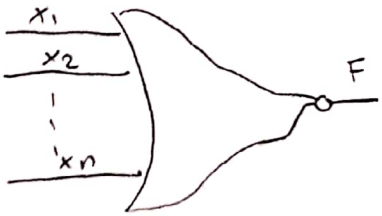
⟶ 

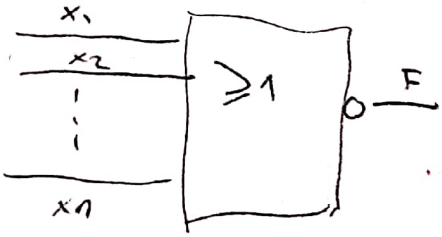
OR ⟶ $F = X_1 + X_2 + X_3 \cdots + X_n$ ⟶ 

⟶ 

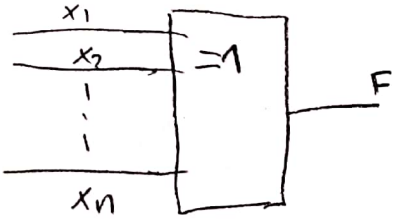
NAND ⟶ $\overline{F} = \overline{(X_1 \cdot X_2 \cdot X_3 \cdots X_n)}$ ⟶ 

⟶ 

NOR ⟶ $\overline{F} = \overline{(X_1 + X_2 + \cdots + X_n)}$ ⟶ 

⟶ 

XOR ⟶ $F = (X_1 \oplus X_2 \oplus X_3 \oplus \cdots \oplus X_n)$ ⟶ 

⟶ 

Yarı Toplayıcı

a	b	Toplam	Elde
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Toplam K-map

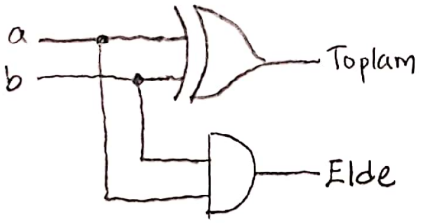
a \ b	0	1
0	0	(1)
1	(1)	0

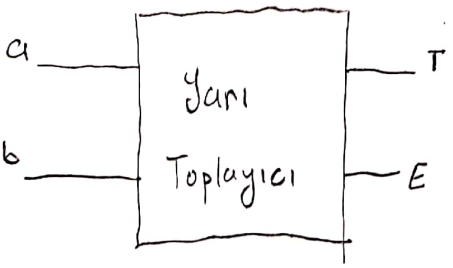
Toplam = $a\bar{b} + \bar{a}b$ (XOR)

Elde K-map

a \ b	0	1
0	0	0
1	0	(1)

Elde = ab (AND)





Tam Toplayıcı

Yarı toplayıcının aksine girişte elde bilgisi vardır.

Eg	a	b	Toplam (T)	Elde (E)
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

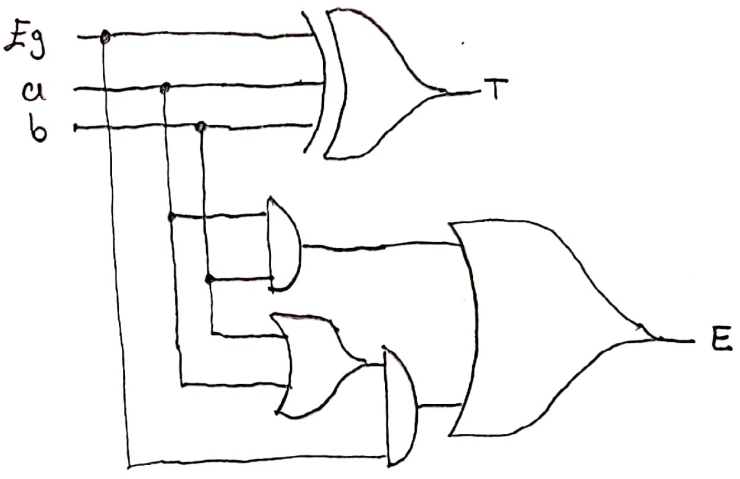
Eg \ ab	00	01	11	10
0	0	1	0	1
1	1	0	1	0

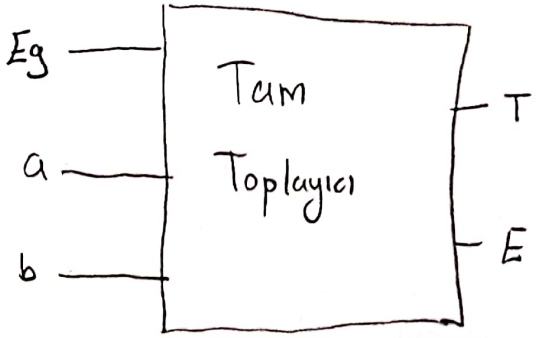
T

$T = Eg \oplus a \oplus b$

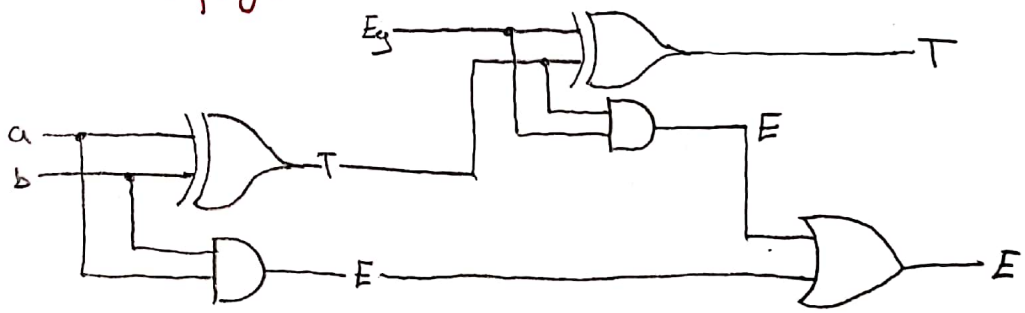
Eg \ ab	00	01	11	10
0	0	0	(1)	0
1	0	(1)	(1)	(1)

$E = ab + Eg(a+b)$





2 yarı toplayıcı ile
1 tam toplayıcı



Yarım Çıkarıcı

a	b	Fark(F)	Borç(B)
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

F

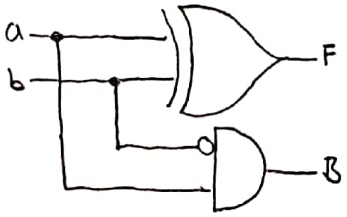
a \ b	0	1
0	0	1
1	1	0

$F = a\bar{b} + \bar{a}b$

B

a \ b	0	1
0	0	1
1	0	0

$B = \bar{a}b$



Tam Çıkarıcı

a	b	Bg	F	B
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

a \ bBg	00	01	11	10
0	0	1	0	1
1	1	0	1	0

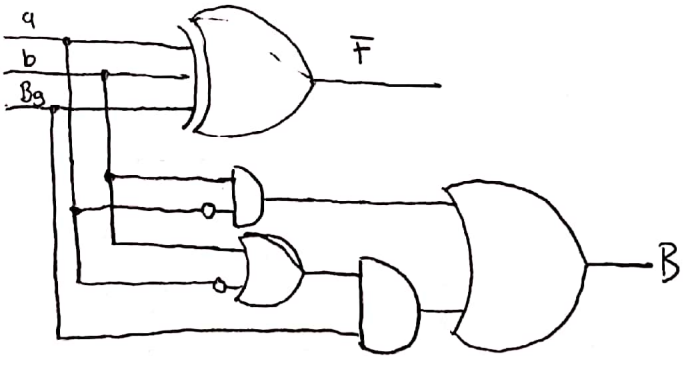
F

$F = a \oplus b \oplus Bg$

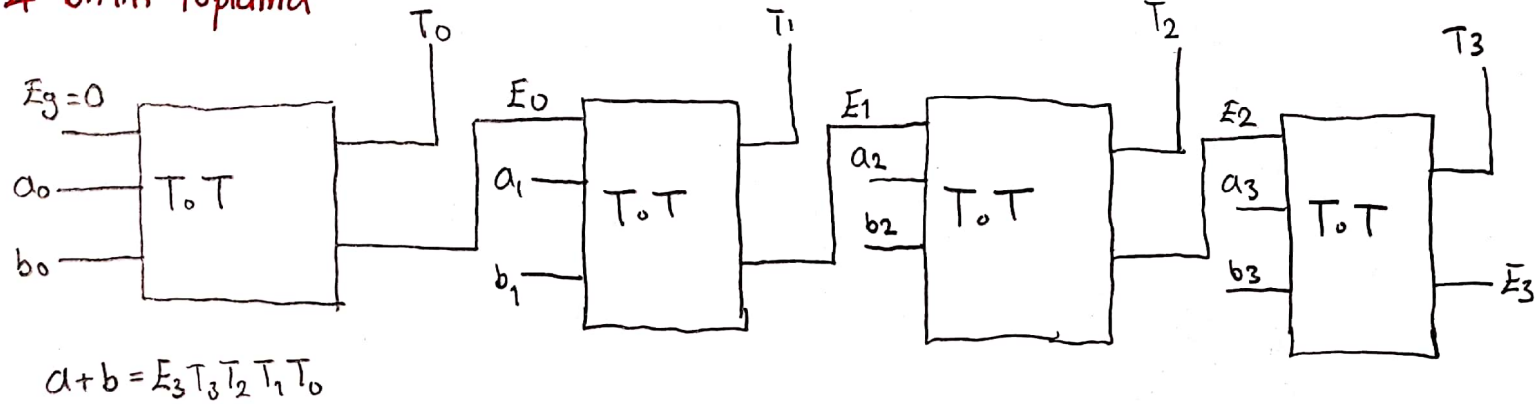
a \ bBg	00	01	11	10
0	0	1	1	1
1	0	0	1	0

B

$B = \bar{a}b + Bg(\bar{a} + b)$

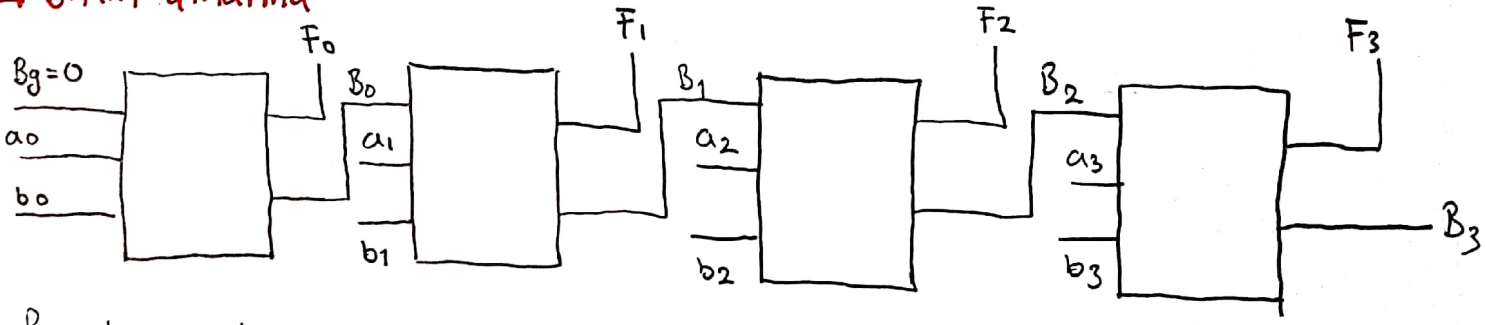


Tam Toplayıcılarla 4 bitlik toplama



$a+b = E_3 T_3 \bar{T}_2 T_1 T_0$

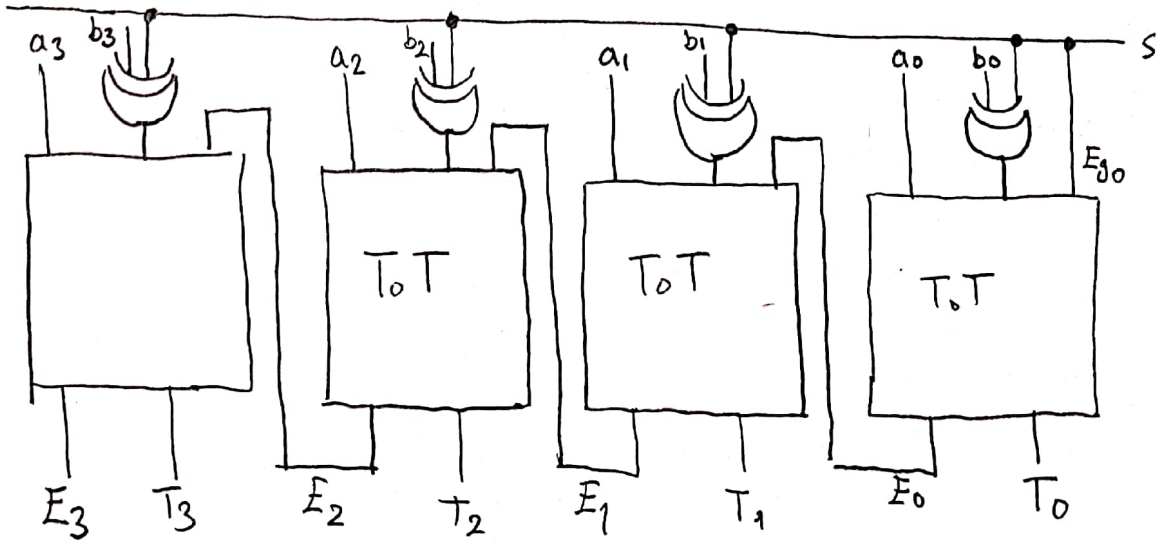
Tam Çıkarıcılarla 4 bitlik çıkarma



B_3 taşma kontrolü için kullanılır.

Toplama/Çıkarma Devresi

Kontrol girişi vasıtasıyla toplama veya çıkarma işlemi seçen ve seçilen işlemi yapan devre.

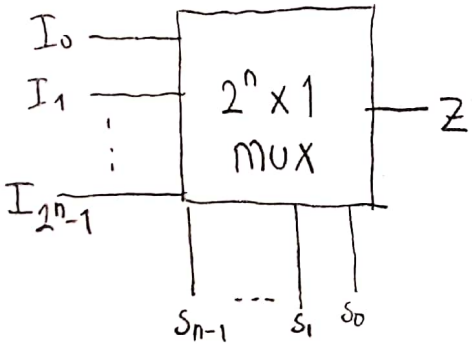


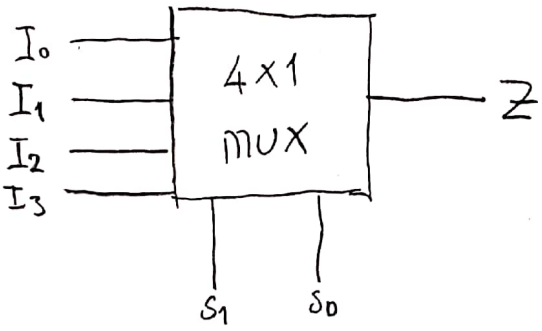
Multiplexer (MUX)

Sayısal veri iletişiminde birçok inputtan tekini seçip outputta gösterir

input sayısı : 2^n

select sayısı : n





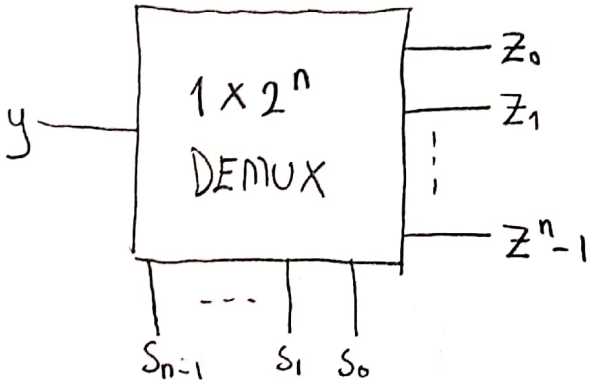
$S_1S_0 = 00 \longrightarrow Z = I_0$

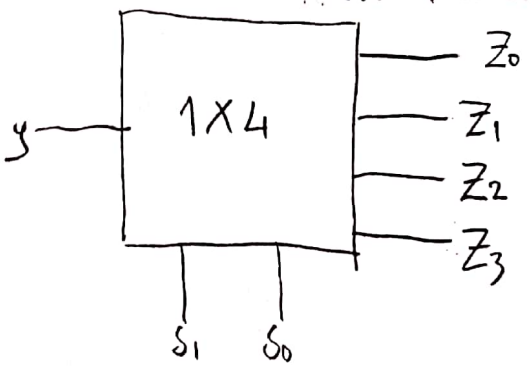
$S_1S_0 = 01 \longrightarrow Z = I_1$

$S_1S_0 = 10 \longrightarrow Z = I_2$

$S_1S_2 = 11 \longrightarrow Z = I_3$

Demultiplexer (DEMUX)





$S_1S_0 = 00 \longrightarrow y = Z_0$

$S_1S_0 = 01 \longrightarrow y = Z_1$

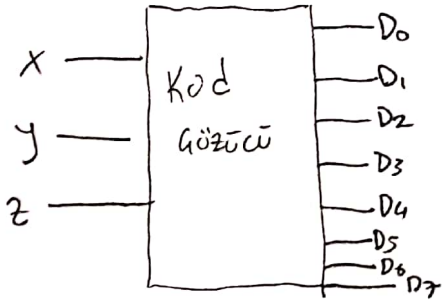
$S_1S_0 = 10 \longrightarrow y = Z_2$

$S_1S_0 = 11 \longrightarrow y = Z_3$

Kod Çözücü

n bitlik bir sözcüğün kodunu çözüp dışarıdaki en çok 2^n çıkış yolundan sadece birini aktif hale getiren kombinezonsal devredir.

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



Kodlayıcı

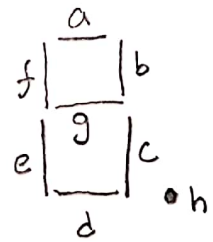
D_3	D_2	D_1	D_0	X	y	hazır
0	0	0	0	Ø	Ø	0
0	0	0	1	0	0	1
0	0	1	Ø	0	1	1
0	1	Ø	Ø	1	0	1
1	Ø	Ø	Ø	1	1	1

$X = D_2 + D_3$

$y = D_3 + D_1 \cdot \overline{D_2}$

$hazır = D_0 + D_1 + D_2 + D_3$

Seven Segment Display

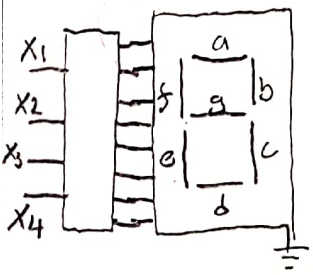


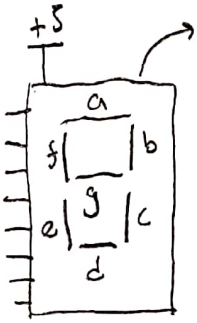
1: bc

2: abged

3: abgcd

1'de yanar





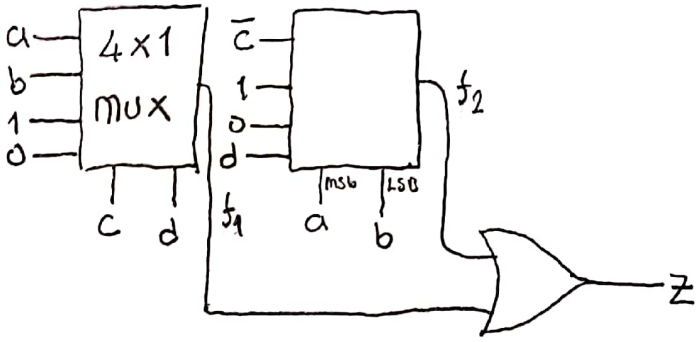
Ø olursa yanar

Ortak anotlu 7segment

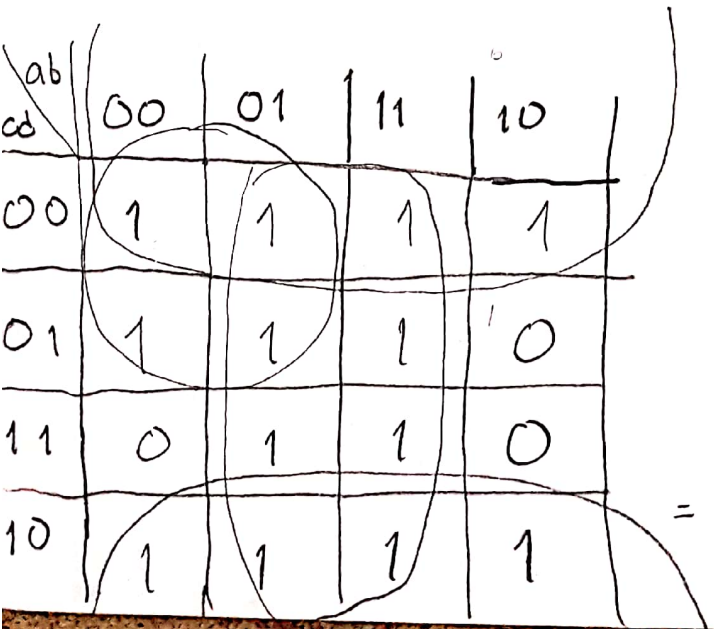
$2^3 < 10 < 2^4$

	X_1	X_2	X_3	X_4	a	b	c	d	e	f	g	h
0:	0	0	0	0	1	1	1	1	1	1	0	0
⋮												
7:	0	1	1	1	1	1	1	0	0	0	0	0

Örnek

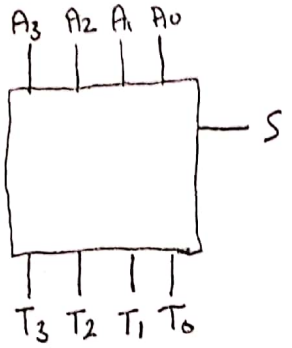


a	b	c	d	f_1	f_2	z
0	0	0	0	0	1	1
0	0	0	1	0	1	1
0	0	1	0	1	0	1
0	0	1	1	0	0	0
0	1	0	0	0	1	1
0	1	0	1	1	1	1
0	1	1	0	1	1	1
0	1	1	1	0	1	1
1	0	0	0	1	0	1
1	0	0	1	0	0	0
1	0	1	0	1	0	1
1	0	1	1	0	0	0
1	1	0	0	1	0	1
1	1	0	1	1	1	1
1	1	1	0	1	0	1
1	1	1	1	0	1	1



$= z = b + \bar{d} + \bar{a}\bar{c}$

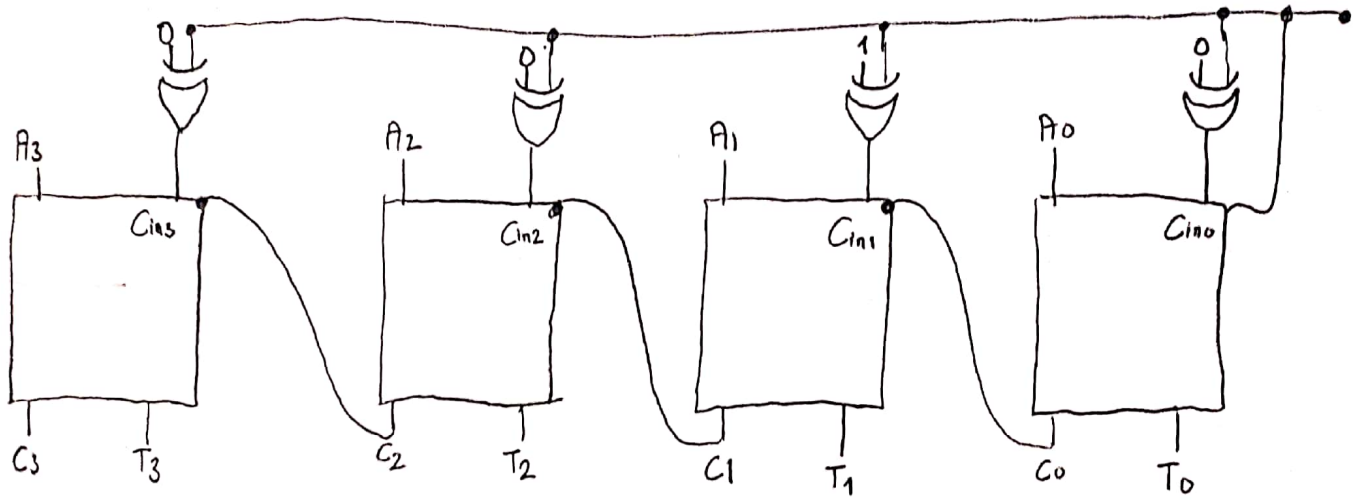
Örnek



Bu devreyi tam toplayıcılarla yapınız.
XOR kullanabilirsiniz.

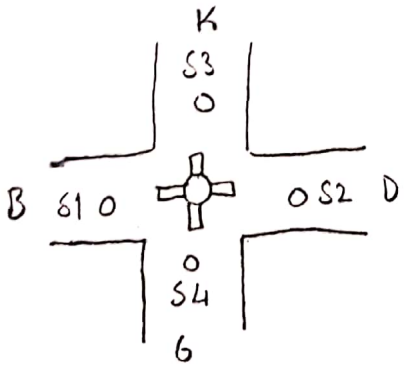
S=0 ⟶ T=A+2

S=1 ⟶ T=A-2



SORU: XOR yerine mux kullansaydık ne olurdu (2x1 mux) + NOT gate kullanılabilir.

Örnek



$\overline{F_{KG}} = F_{DB}$

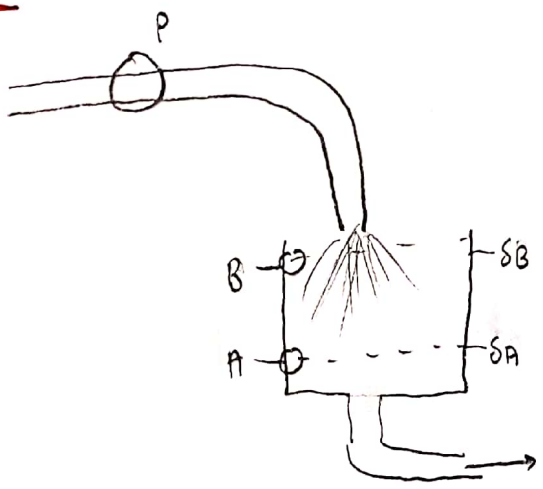
Trafik lambaları ve yollar

- $S_1 S_2 = 11 \longrightarrow S_3 S_4 = \emptyset\emptyset \longrightarrow F_{DB} = 1$ → yeşil ışık
- $S_1 S_2 = 10 \cup 01 \longrightarrow S_3 S_4 = 10 \cup 01 \longrightarrow F_{DB} = 1$
- $S_1 S_2 S_3 S_4 \longrightarrow F_{DB} = 1$
- Yukarıdakiler hariç $F_{KG} = 1$

Karnaugh ile gerçekle

S1	S2	S3	S4	F_{DB}	F_{KG}
0	0	0	0	1	0
0	0	0	1	0	1
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	1	0
0	1	0	1	1	0
0	1	1	0	1	0
0	1	1	1	0	1
1	0	0	0	1	0
1	0	0	1	1	0
1	0	1	0	1	0
1	0	1	1	0	1
1	1	0	0	1	0
1	1	0	1	1	0
1	1	1	0	1	0
1	1	1	1	1	0

örnek



A < Su < B olmalı

- Su seviyesi maksimum, pompa çalışıyorsa, pompa kapat.
- Su seviyesi maksimum, pompa duruyor, pompa kapalı.
- Su seviyesi minimum, pompa duruyor, pompa çalıştırılacak.
- Su seviyesi minimum, pompa çalışıyor, pompa çalışıyor.

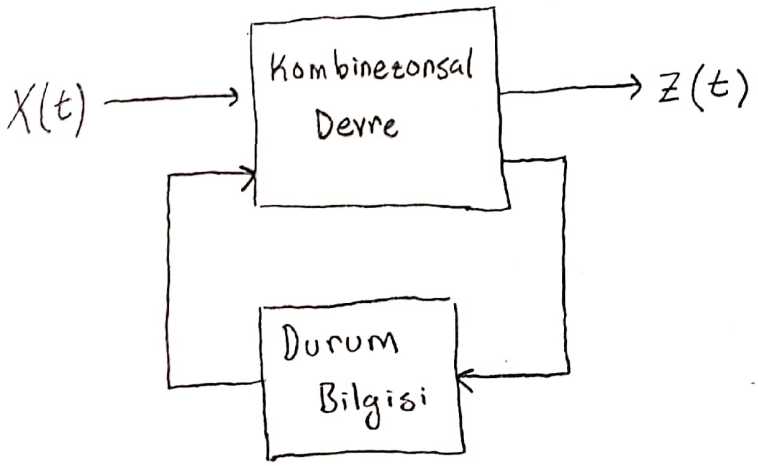
AB=01 ⟶ D=0 ⟶ F=0

AB=01 ⟶ D=1 ⟶ F=1

A	B	D	F
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	Mümkün değil
1	0	1	
1	1	0	0
1	1	1	0

A \ BD	00	01	11	10
0	1	1	1	0
1			0	0

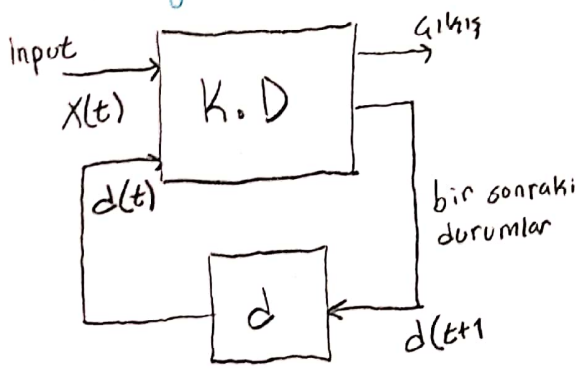
Ardışıl Devre



Mealy makinalarında t anındaki çıkış değeri t anındaki durumlara ve t anındaki girişlere bağlıdır.

Moore makinalarında ise t anındaki çıkış değeri yalnızca t anındaki durumlara bağlıdır.

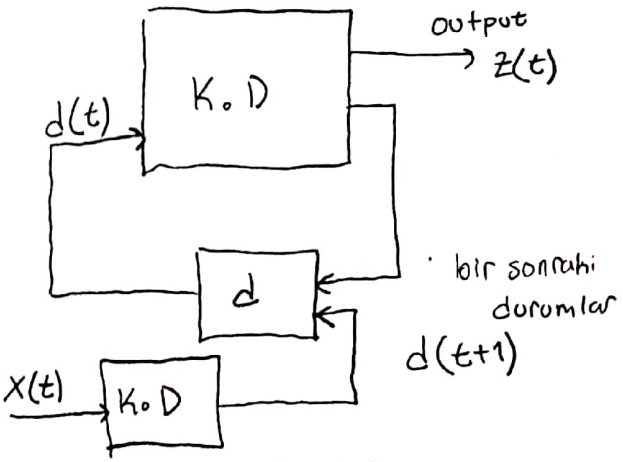
Mealy Makinesi



$Z(t) = f(d(t), X(t))$

$d(t+1) = f(d(t), x(t))$

Moore Makinesi

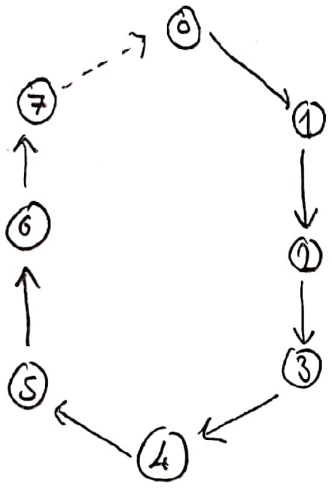


$Z(t) = f(d(t))$

$d(t+1) = f(d(t), X(t))$

Durumlar ve Durum Diyagramı

0'dan 7'ye sayıcının durumları



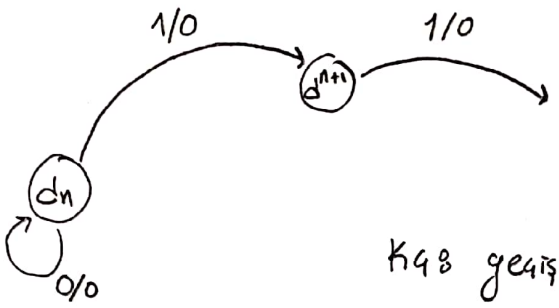
Durum Tablosu

Örnek: tek sayılarda çıkışı 1 yapan sayıcı durum tablosu.

Şimdiki Durumlar			Bir Sonraki Durumlar			Çıkış
A	B	C	A^{t+1}	B^{t+1}	C^{t+1}	Z
0	0	0	0	0	1	1
0	0	1	0	1	0	0
0	1	0	0	1	1	1
0	1	1	1	0	0	0
1	0	0	1	0	1	1
1	0	1	1	1	0	0
1	1	0	1	1	1	1
1	1	1	0	0	0	0

Durum Diyagramı

Ardışıl devrenin durumlarını ve durumlar arası geçişi gösteren bir graftır.



Kaç geçiş koşulu / çıkış değeri

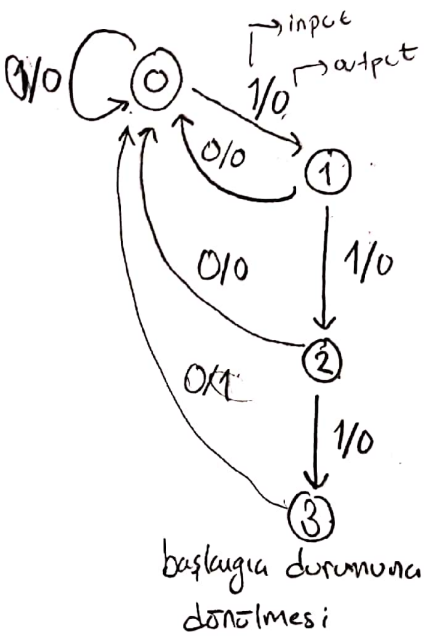
→ çıkış 0

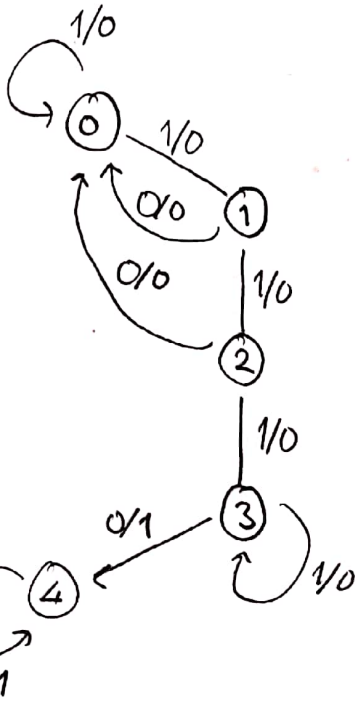
→ girişin lojik 1 olduğunu belirtir

Örnek

Şifre devresi tasarımı

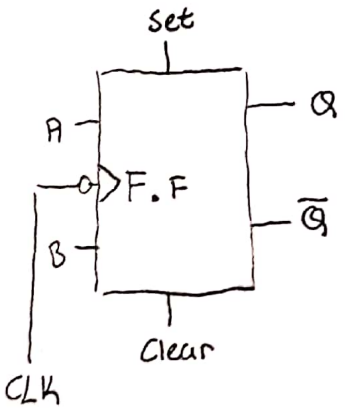
devrenin 1 girişi ve 1 çıkışı vardır.
Kendisine girilen şifre koduna göre çıkışı lojik 1 yapması isteniyor.
Şifrenin lojik değerde girildiği ve 1110 olduğu varsayılmaktadır. Yani art arda 3 tane lojik 1 ve daha sonra lojik 0 girildiğinde çıkışın 1; aksi durumda çıkışın 0 olması isteniyor. Devrenin olabilecek durumlarını belirleyiniz ve durum diyagramını çiziniz.





Flip-Floplar

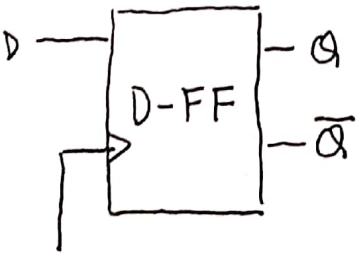
Flip floplar 1 bitlik saklama birimleridir. En yalın saklama elemanı olarak tek başına veya birden çok flip flop bir araya getirilip; saklayıcı, sayıcı, bellek vb. birimler oluşturulabilir. Mikroişlemci üzerinde veya kartların üzerinde bayrak olarak olur. Değişkenler de flip flop üzerinde tutulur.



D-FF (Delay)

$Q(t+1) = D(t)$

$Y = D$



Giriş	CLK	Çıkışlar Q	$\overline{Q}$
0	↓	0	1
1	↓	1	0

$Q \rightarrow y$

$Q^{t+1} \rightarrow Y$

Uyarma tablosu:

$Q\ Q^{t+1}$ → çıkış $y\ Y$	00	01	11	10
D	0	1	1	0

↓

y \ D	0	1
0	0	1
1	0	1

$Y = D$

SR FF (Set-Reset)

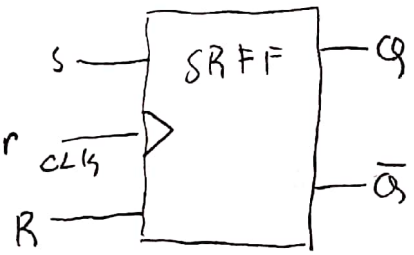
A) $Q(t+1) = Q(t)\cdot\overline{R(t)} + S(t)$; $R(t)\cdot S(t) = 0$, $R(t)\cdot S(t) = 1$

B)

Giriş		CLK	Çıkışlar	
S	R		Q^{t+1}	$\overline{Q}^{t+1}$
0	0	⎍	Q	$\overline{Q}$
0	1	⎍	0	1
1	0	⎍	1	0
1	1	⎍	tanımsız	

Set Reset

→ Çıkış Korunuyor



C)

yY	00	01	10	11
SR	0Ø	10	Ø0	01

→ Çıkış

K-map

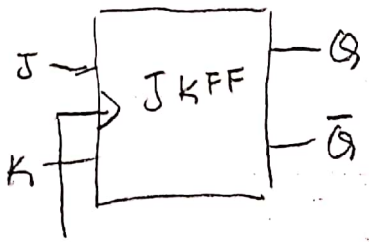
y \ SR	00	01	11	10
0	0	0	K	1
1	1	0	K	1

K = 1

$Y = S + \overline{R}y$

JK FF

$Q(t+1) = Q(t)\cdot\overline{K(t)} + Q(t)\cdot J(t)$



Giriş		CLK	Çıkışlar	
J	K		Q	$\overline{Q}$
0	0	⎍	Q	$\overline{Q}$
0	1	⎍	0	1
1	0	⎍	1	0
1	1	⎍	$\overline{Q}$	0
↓ R	↓ S			

Q Q^{t+1}

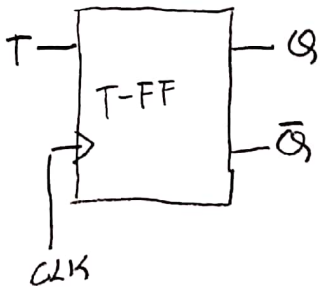
yY	00	01	11	10
Jk	0Ø	1Ø	Ø0	Ø1

y \ JK	00	01	11	10
0	0	0	1	1
1	1	0	0	1

$Y = J\overline{y} + \overline{K}y$

T-FF (Toggle)

$Q(t+1) = Q(t) \oplus T(t)$



Giriş	CLK	Çıkış
T		Q_t $\bar{Q}_t$
0	⎍	Q_t $\bar{Q}_t$
1	⎍	$\bar{Q}_t$ Q_t

yY	00	01	11	10
T	0	1	0	1

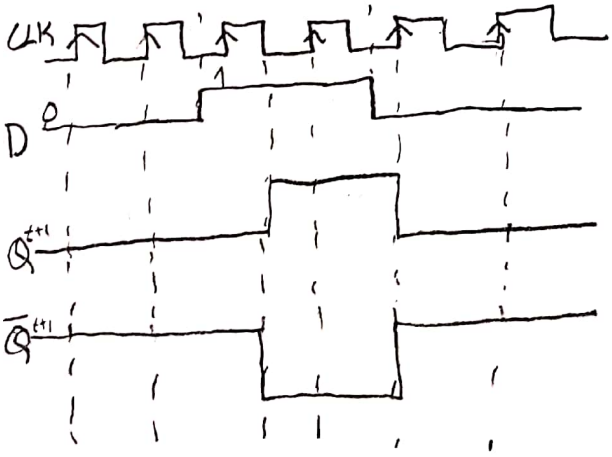
↓

K-map

y \ T	0	1
0	0	1
1	1	0

$= y \oplus T$

Örnek



SRFF

Q^t (önceki durum)	S	R	Q^{t+1} (sonraki durum)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	K
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	K
	↓ Set	↓ Reset	

S=R=0 ise önceki durum korunur

S=R=1 ise kalmsız

JKFF

Q^t (önceki durum)	J	K	Q^{t+1} (sonraki durum)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0
	↓ Set	↓ Reset	

mantığı

J=K=0 ise önceki durum korunur

J=K=1 ise önceki durumun complementi Q^{t+1}'de görülür.

1/2

4—

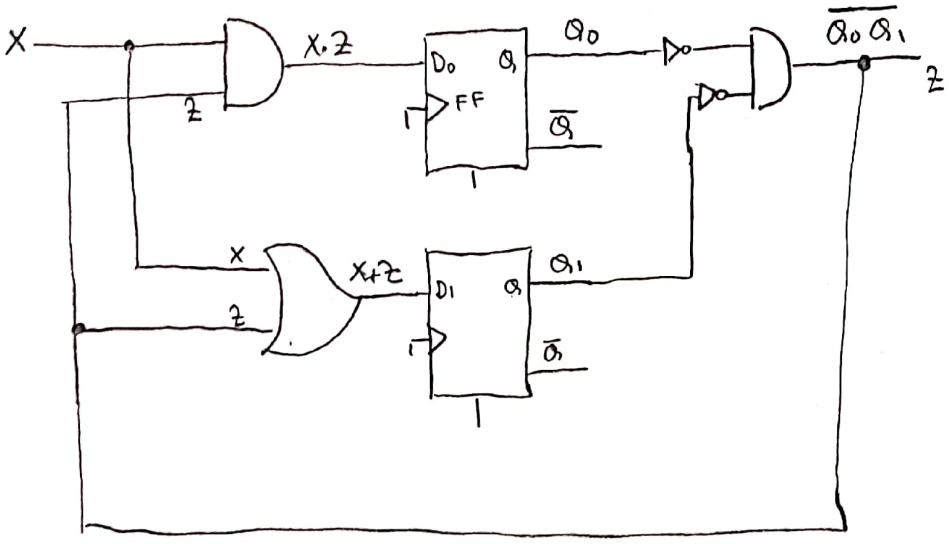
DFF

Q^t	D	Q^{t+1}
0	0	0
0	1	1
1	0	0
1	1	1

TFF

Q^T	T	Q^{t+1}	
0	0	0	0 etkisiz
0	1	1	
1	0	1	1 → not alır
1	1	0	

Ardışıl Devre Analiz Yöntemi



DFF

Q^t	D	Q^{t+1}
0	0	0
0	1	1
1	0	0
1	1	1

$$Q^{t+1} = D$$

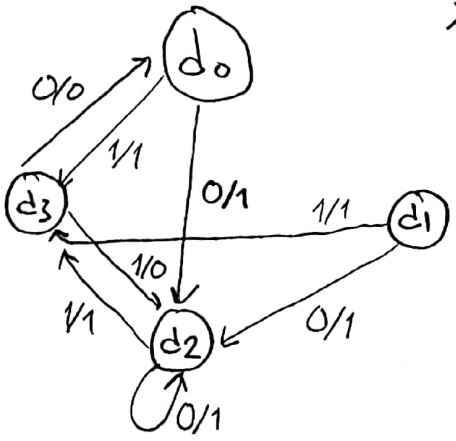
$$D_0 = X \cdot Z = X \cdot (\overline{Q_0^t \cdot Q_1^t})$$

$$D_1 = X + Z = X + (\overline{Q_0^t \cdot Q_1^t})$$

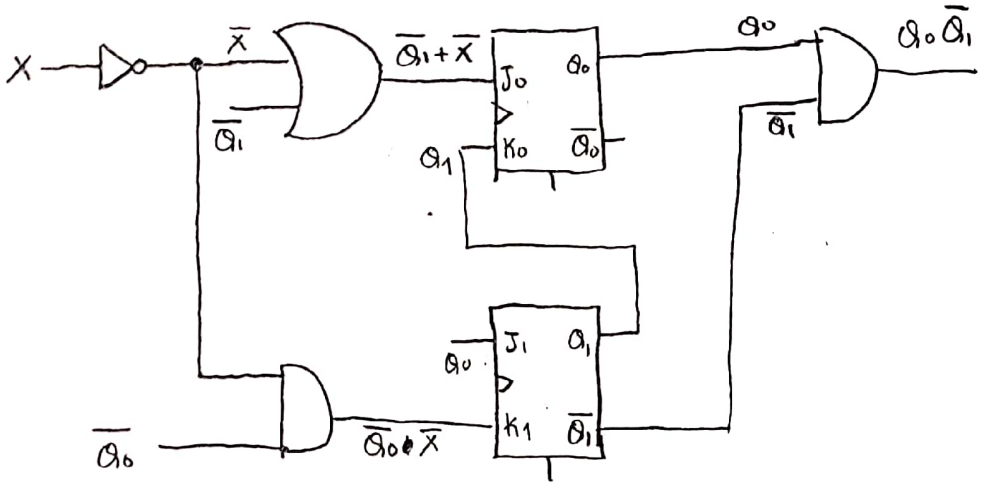
$$Z = \overline{Q_0 \cdot Q_1}$$

Simgesel durumlar	şimdiki durumlar Q_1^t	Q_0^t	flip flop inputs X=0 D_1D_0	X=1 D_1D_0	bir sonraki durumlar X=0 $Q_1^{t+1}Q_0^{t+1}$	X=1 $Q_1^{t+1}Q_0^{t+1}$	çıkış Z
d_0	0	0	10	11	10	11	1
d_1	0	1	10	11	10.	11	1
d_2	1	0	10	11	10	11	1
d_3	1	1	00	10	00	00	0

Şimdiki durumlar	bir sonraki durumlar X=0	X=1	Z
d_0	d_2	d_3	1
d_1	d_2	d_3	1
d_2	d_2	d_3	1
d_3	d_0	d_2	0



X/Z → i/o



JKFF
↓ ↓
set reset

00 → önceki durum koru
11 → komplement al

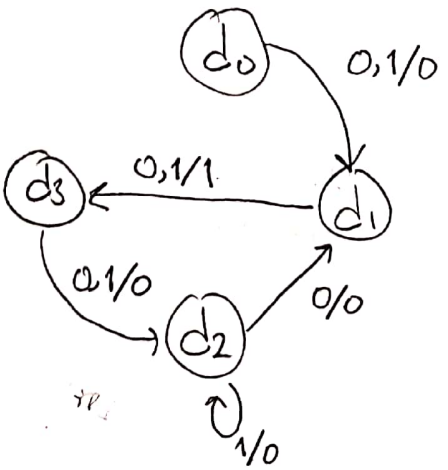
$J_0 = \bar{Q}_1 + \bar{X}$ $K_0 = Q_1$

$J_1 = Q_0$ $K_1 = \bar{X} \bullet \bar{Q}_0$

$Z = \bar{Q}_0 \cdot \bar{Q}_1$

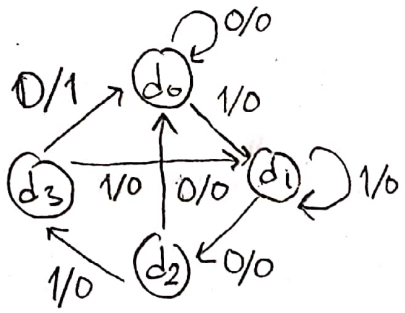
Simgesel durum adları	şimdiki durumlar Q_1	Q_0	flip flop inputs X=0 J_1K_1	J_0K_0	X=1 J_1K_1	J_0K_0	bir sonraki durumlar X=0 Q_1Q_0	X=1 Q_1Q_0	Çıkış Z $Q_0\cdot\bar{Q}_1$
d_0	0	0	01	10	00	10	01	01	0
d_1	0	1	11	10	10	10	11	11	1
d_2	1	0	01	11	00	01	01	10	0
d_3	1	1	11	11	10	01	10	10	0

Sonraki durum b_0 X=0	δ_0 X=1	z
d_0	d_1	d_1	0
d_1	d_3	d_3	1
d_2	d_1	d_2	0
d_3	d_2	d_2	0



Ardışıl Devre Tasarım Yöntemi

Girişi X, çıkışı Z ile ifade edilen ardışıl devrenin girişine 1010 bit dizisi geldiğinde çıkışı lojik 1 olan, aksi durumda lojik 0 yapan devreyi DFF ile yap



0110

0101

1011

1011101

1010

110

100 100

10110

Şimdiki durum	bir sonraki durumlar X=0	X=1	Z X=0	X=1
d_0	d_0	d_1	0	0
d_1	d_2	d_1	0	0
d_2	d_0	d_3	0	0
d_3	d_0	d_1	1	0

Simgesel durum adları	Şimdiki durumlar Q_1^t	Q_0^t	Bir sonraki Durumlar X=0 $Q_1^{t+1}Q_0^{t+1}$	X=1 $Q_1^{t+1}Q_0^{t+1}$	Flip Flop Inputs X=0 D_1D_0	X=1 D_1D_0	Z X=0	X=1
d_0	0	0	00	01	00	01	0	0
d_1	0	1	10	01	10	01	0	0
d_2	1	0	00	11	00	11	0	0
d_3	1	1	00	01	00	01	1	0

D_1

X \ Q_1Q_0	00	01	11	10
0	0	1	0	0
1	0	0	0	1

$D_1 = \overline{X}Q_1\overline{Q_0} + XQ_0\overline{Q_1}$

D_0

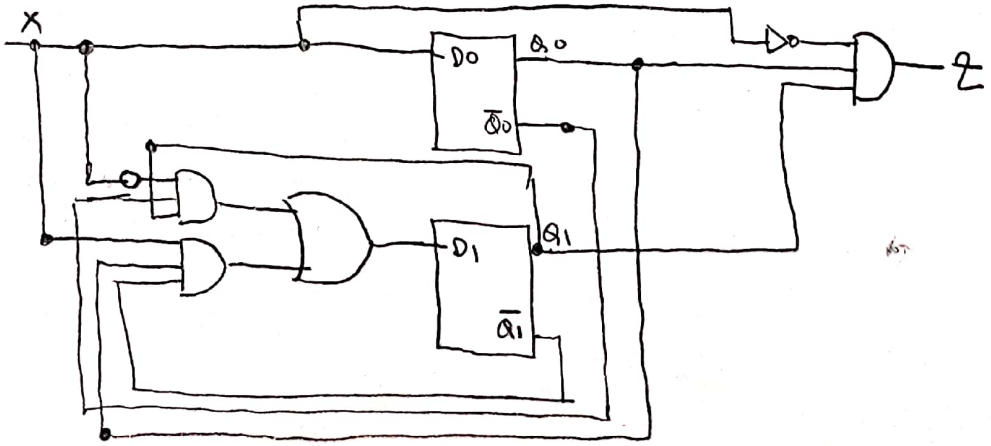
X \ Q_1Q_0	00	01	11	10
0	0	0	0	0
1	1	1	1	1

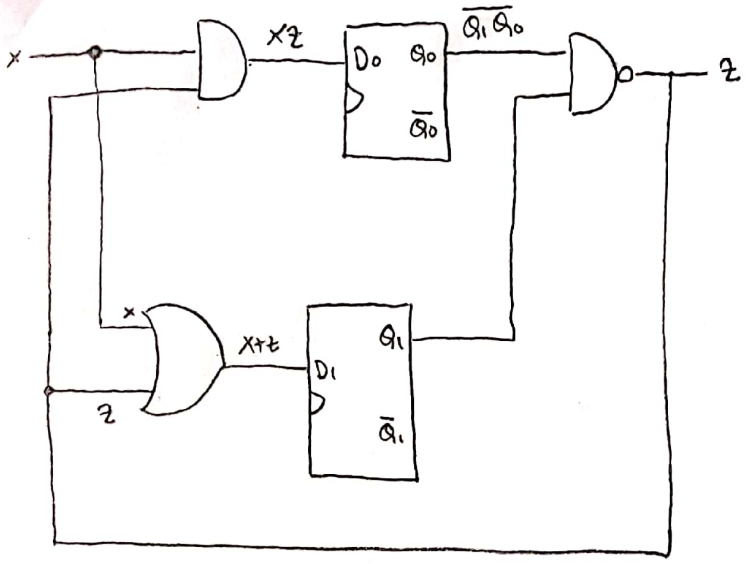
$D_0 = X$

Z

X \ Q_1Q_0	00	01	11	10
0	0	0	1	0
1	0	0	0	0

$Z = \overline{X}Q_1Q_0$





$D_0 = X \cdot (\overline{Q_1 Q_0})$

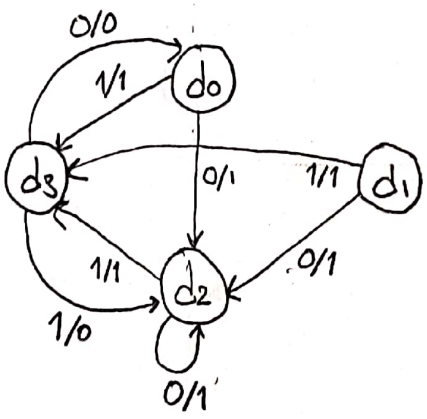
$D_1 = X + (\overline{Q_1 Q_0})$

$D_0 = Xz \qquad z = \overline{Q_1 Q_0}$

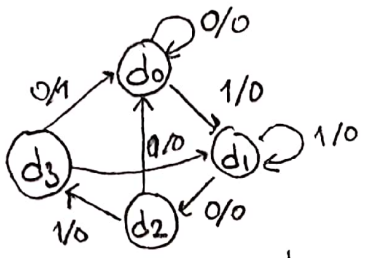
$D_1 = X + y$

durum	şu anki durum ① X=0 Q_1^t	X=1 Q_0^t	bir sonraki durum ③ X=0 $Q_1^+ Q_0^+$	X=1	flip flop inputs ② X=0 $D_1 D_0$	X=1 $D_1 D_0$	z x=0	x=1
d_0	0	0	10	11	10	11	1	
d_1	0	1	10	11	10	11	1	
d_2	1	0	10	11	10	11	1	
d_3	1	1	00	10	00	10	0	

d	δ d X=0	X=1	output
d_0	d_2	d_3	1
d_1	d_2	d_3	1
d_2	d_2	d_3	1
d_3	d_0	d_2	0



Girişi X, çıkışı Z, fsa, şifre: 1010 DFF



110

1011010

10

100

Simgesel	Şu anki durum Q_1^t	Q_0^t	sonraki durum X=0 $Q_1^+Q_0^+$	X=1 Q_1Q_0	DFF X=0 D_1D_0	X=1 D_1D_0	Z X=0	X=1
d0	0	0	00	01	00	01	0	0
d1	0	1	10	01	10	01	0	0
d2	1	0	00	11	00	11	0	0
d3	1	1	00	01	00	01	1	0

$D_1=?$ $D_0=?$ $Z=?$

Q_1Q_0 / X	00	01	11	10
0	0	(1)	0	0
1	0	0	0	(1)

D_1

$D_1 = \bar{X}\cdot\bar{Q}_1Q_0 + XQ_1\bar{Q}_0$

Q_1Q_0 / X	00	01	11	10
0	0	0	0	0
1	(1	1	1	1)

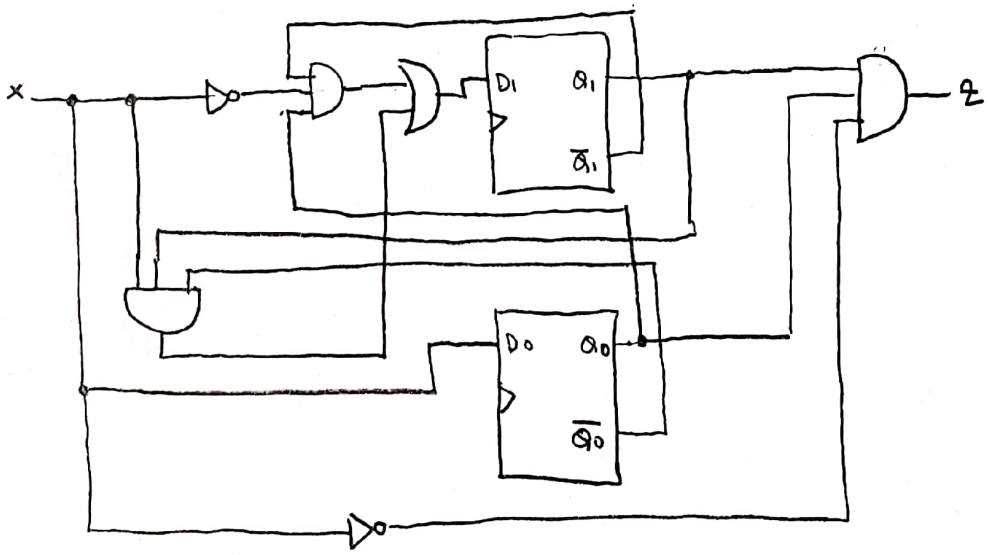
D_0

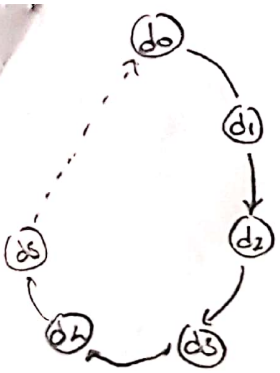
$D_0 = X$

Q_1Q_0 / X	00	01	11	10
0	0	0	(1)	0
1	0	0	0	0

Z

$Z = \bar{X}Q_1Q_0$





JK FF → 00 → aynı
11 → tümleyen
) set reset

simgesel	şimdiki durumlar Q_2 Q_1 Q_0	Bir sonraki durumlar $Q_2^+ Q_1^+ Q_0^+$	JKFF J_2K_2	J_1K_1	J_0K_0
d_0	0 0 0	0 0 1	0Ø	0Ø	1Ø
d_1	0 0 1	0 1 0	0Ø	1Ø	Ø1
d_2	0 1 0	0 1 1	0Ø	Ø0	1Ø
d_3	0 1 1	1 0 0	1Ø	Ø1	Ø1
d_4	1 0 0	1 0 1	Ø0	0Ø	1Ø
d_5	1 0 1	1 1 0	Ø0	1Ø	Ø1
d_6	1 1 0	1 1 1	Ø0	Ø0	1Ø
d_7	1 1 1	0 0 0	Ø1	Ø1	Ø1

$J_1 = ?$ $K_1 = ?$
$J_0 = ?$ $K_0 = ?$
$J_2 = ?$ $K_2 = ?$

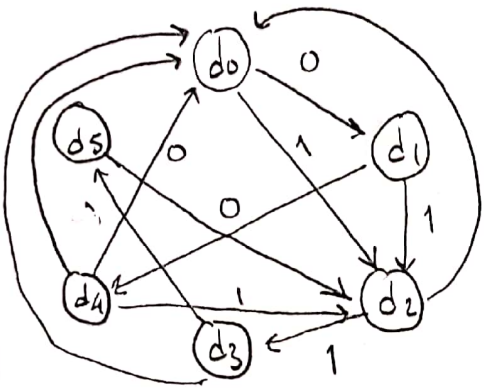
x=0 ise $\{0,1,4\}$

x=1 ise $\{2,3,5\}$

00 → aynı

11 → değil

S/R



00 → 0Ø

01 → 1Ø

10 → Ø1

11 → Ø0

S	Q_2	Q_1	Q_0	δd x=0 $Q_2Q_1Q_0$	δd x=1 $Q_2Q_1Q_0$	x=0 J_2K_2	J_1K_1	J_0K_0	x=1 J_2K_2	J_1K_1	J_0K_0
d0	0	0	0	001	010	0Ø	0Ø	1Ø	0Ø	1Ø	0Ø
d1	0	0	1	100	010	1Ø	0Ø	Ø1	0Ø	1Ø	Ø1
d2	0	1	0	011	000	0Ø	Ø0	1Ø	0Ø	Ø1	0Ø
d3	0	1	1	101	000	1Ø	Ø1	Ø0	0Ø	Ø1	Ø1
d4	1	0	0	000	010	Ø1	0Ø	0Ø	0Ø	1Ø	0Ø
d5	1	0	1	010	000	Ø1	1Ø	Ø1	0Ø	0Ø	Ø1

1

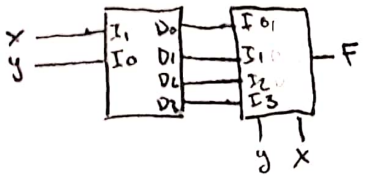
$F = a'b + ab'c'$ $\quad a=b=1$ olmuyor

a	b	c	f
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	∅
1	1	1	∅

c \ ab	00	01	11	10
0	0	1	∅	1
1	0	1	∅	0

$f = b$

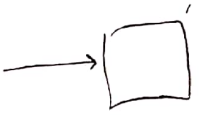
2



x	y	f
0	0	1
0	1	0
1	0	0
1	1	1

x \ y	0	1
0	1	0
1	0	1

$f = x'y' + xy$



parity bit $1 \rightarrow$ gift $\rightarrow 1$
$1 \rightarrow$ tek $\rightarrow 0$

S_d	δ_d	δ_d		DFF		z	
		x=0	x=1	x=0	x=1	x=0	x=1
	Q_0	Q_0	Q_0	D_0	D_0		
d_0	0	0	1	0	1	1	0
d_1	1	1	0	1	0	0	1

Q_0 \ x	0	1
0	0	1
1	1	0

D_0

$D_0 = x \cdot \bar{Q}_0 + \bar{x} \cdot Q_0 = x \,(xor)\, Q_0$

Q_0 \ x	0	1
0	1	0
1	0	1

z

$z = Q \,(xor)\, x$

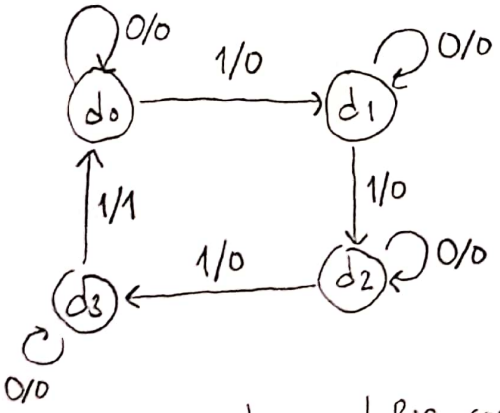
Bir havuz ve havuz kenarında atlama platformu var. Platform 3 kişi taşıyor 1 kişi daha eklenirse eğiliyor düşüyorlar geri eski haline geliyor.

Devrenin tek girişi P.

P=0 iken kişi sayısı değişmiyor.

P=1 iken 1 kişi daha ekleniyor platforma.

Devrenin çıkışı F'dir. F=0 ise platform normal durumda, F=1 ise platform düşme durumunda. Devreyi JK ile gerçekleştiriz.



	Şimdiki Durum Q_1	Q_0	P	Bir sonraki Durumlar Q_1^+	Q_0	$J_1 K_1$	$J_0 K_0$	F
d0	0	0	0	0	0	0 Ø	0 Ø	0
d0	0	0	1	0	1	0 Ø	1 Ø	0
d1	0	1	0	0	1	0 Ø	Ø 0	0
d1	0	1	1	1	0	1 Ø	Ø 1	0
d2	1	0	0	1	0	Ø 0	0 Ø	0
d2	1	0	1	1	1	Ø 0	1 Ø	0
d3	1	1	0	1	1	Ø 0	Ø 0	0
d3	1	1	1	0	0	Ø 1	Ø 1	1

P \ Q_1Q_0	00	01	11	10
0	0	0	Ø	Ø
1	0	1	Ø	Ø

$J_1 = Q_0 P$

P \ Q_1Q_0	00	01	11	10
0	Ø	Ø	0	0
1	Ø	Ø	1	0

$K_1 = Q_0 P$

P \ Q_1Q_0	00	01	11	10
0	0	Ø	Ø	0
1	1	Ø	Ø	1

$J_0 = P$

P \ Q_1Q_0	00	01	11	10
0	Ø	0	0	Ø
1	Ø	1	1	Ø

$K_0 = P$

P \ Q_1Q_0	00	01	11	10
0	0	0	0	0
1	0	0	1	0

$F = P Q_1 Q_0$