

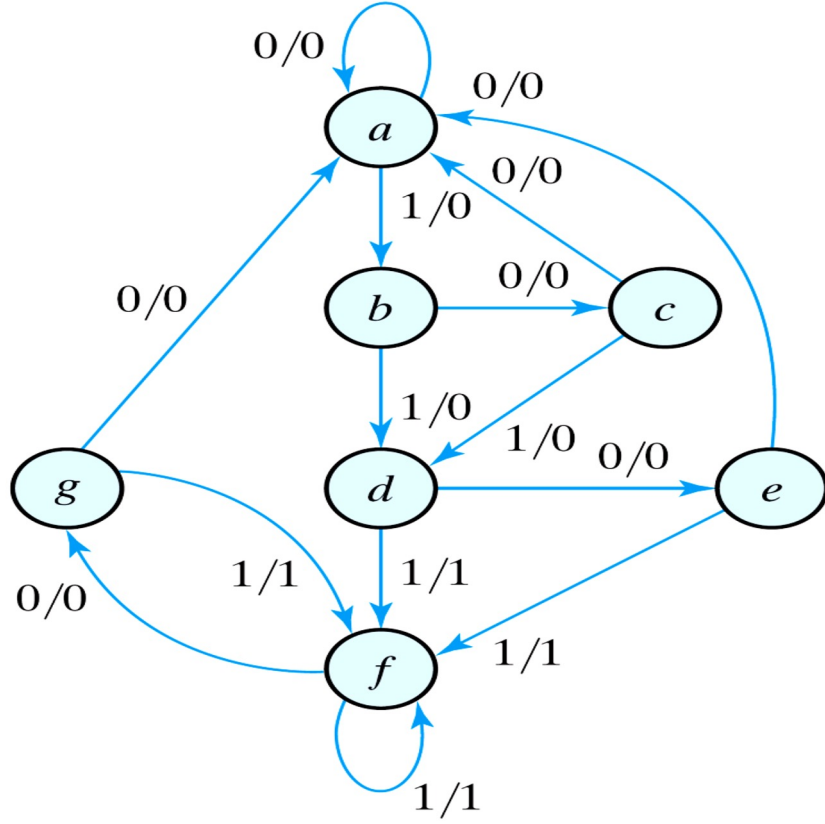
Lojik Devreler

Doç. Dr. Fatih Evran
Elektrik ve Elektronik Mühendisliği
Düzce Üniversitesi

Ardışıl Devrelerde Durum İndirgenmesi ve Durum Kodlaması

Ardışıl lojik devrelerin tasarımı sırasında oluşturulan bazı durum tabloları veya durum diyagramları incelendiğinde görüŖe dayalı olarak bazı Ŗimdiki durumların aynı giriŖ deęerleri için aynı çıkıŖ deęerlerini verdięi ve sonraki durumlarda da aynı olduęu görülebilir. Bu durum tablosunda aynı durumlar için farklı kodlar verilmiŖ demektir. Bunun sonucunda aynı olan durumlara aynı kod verilerek durum tablosu indirgenebilir. Böylece bazı durumlarda daha az sayıda FF ile geręekleŖtirme yapılabilir.

Örnek : Aşağıda verilen durum diyagramını satır eşleştirme tablosu yöntemiyle indirgeyiniz.



Durum diyagramı

İlk aşamada devrenin durum/çıkış tablosu hazırlanır:

Şimdiki durum	Gelecek durum		Çıkış	
	x=0	x=1	x=0	x=1
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	g	f	0	1
g	a	f	0	1

Durumların azaltılabilmesi için, şimdiki duruma ait sonraki durumun ve çıkışın eşit olması gerekir. Bu özelliği sağlayan durumlar “e” ve “g” dir. Tabloda “g” yerine “e” yazılır ve “g” durumu atılır.

Şimdiki durum	Gelecek durum		Çıkış	
	x=0	x=1	x=0	x=1
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	g	f	0	1
g	a	f	0	1

$e = g$



Şimdiki durum	Gelecek durum		Çıkış	
	x=0	x=1	x=0	x=1
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	e	f	0	1

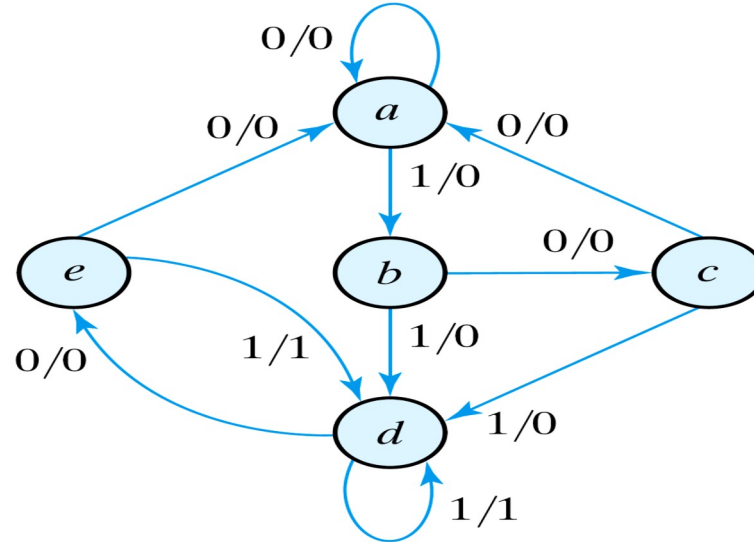
Tablo incelendiğinde “ d ” ve “ f ” durumlarının gelecek durumları ve çıkışları aynı olduğu için, tabloda “ f ” durumu silinip, “ f ” yerine “ d ” yazılabilir.

Şimdiki durum	Gelecek durum		Çıkış	
	x=0	x=1	x=0	x=1
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	e	f	0	1

$d = f$



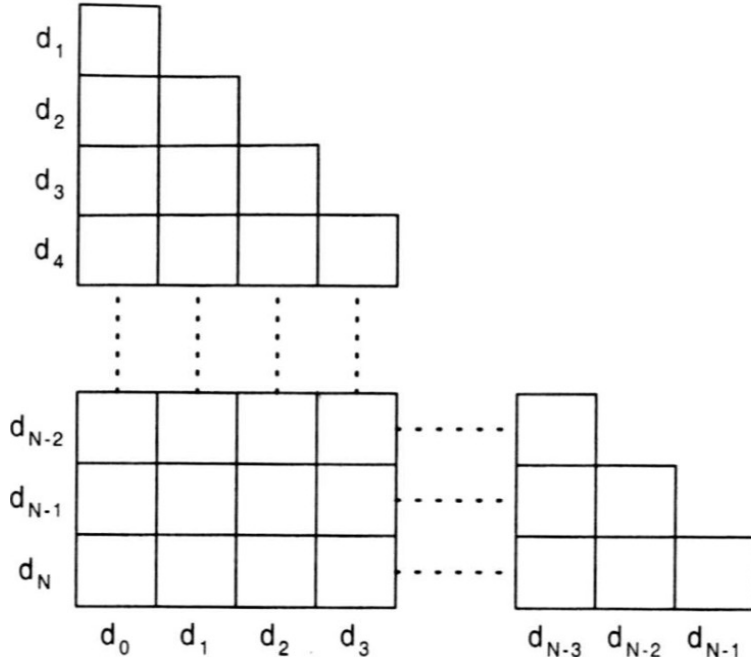
Şimdiki durum	Gelecek durum		Çıkış	
	x=0	x=1	x=0	x=1
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	d	0	1
e	a	d	0	1



Azaltılmış durum diyagramı

Eşleştirme Tablosu ile Durum İndirgemesi Yöntemi

Durum indirgemesi, durum tablosuna bakılarak görüşe dayalı olarak yapılabilir. Fakat karmaşık devrelerde durum indirgemesinin görüşe dayalı yapılmasının zor olması nedeniyle “Eşdeğerlik tablosu” yöntemi kullanılır.



N durumlu bir ardışıl devrenin eşdeğerlik tablosu

- İlk olarak tamamen farklı olan durumlardan başlanarak bunların kesiştiği yerler $\times$ olarak işaretlenir.
- Tablonun satır ve sütunlarının kesiştiği kutuya, durumlar farklıysa $\times$ aynıysa $\checkmark$ işareti konur.
- Sonra d_0 durumuyla en üstteki d_1 durumundan başlanarak farklı olan durumlar kutuya koşul olarak yazılarak aşağıya doğru devam edilir.
- d_N durumu değerlendirildikten sonra bir sonraki d_1 sütununa geçilerek bütün sütunlar bitene kadar işleme devam edilir.

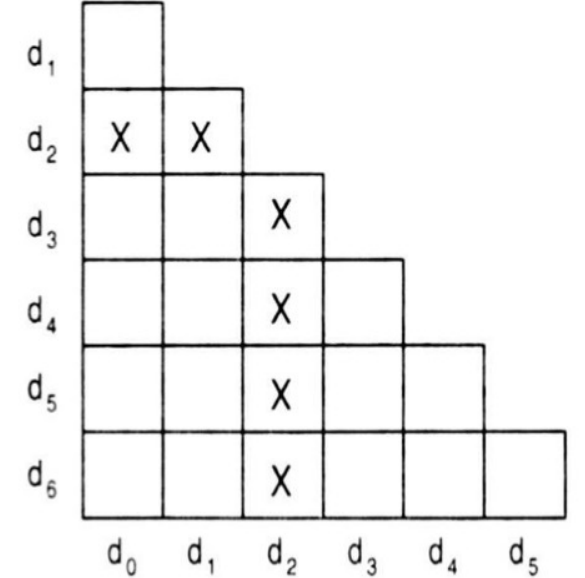
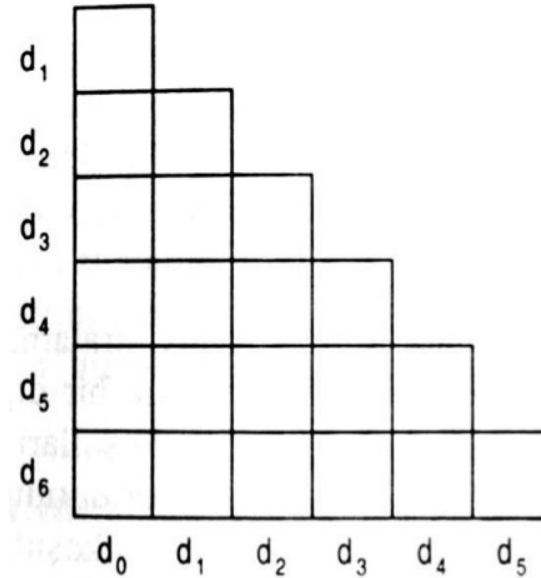
- Koşul yazılan kutular sorgulanarak eşdeğer olan ve olmayan durumlar belirlenir.
- Son olarak durum tablosunda eşdeğer durumlara aynı durum kodu verilerek indirgeme yapılır.

Örnek : Aşağıda verilen durum tablosunu eşleştirme tablosu yöntemiyle indirgeyiniz.

Şimdiki durum	Gelecek durum		Çıkış	
	x=0	x=1	x=0	x=1
d_0	d_1	d_0	0	0
d_1	d_2	d_4	1	0
d_2	d_5	d_2	1	1
d_3	d_4	d_0	0	0
d_4	d_2	d_4	1	0
d_5	d_4	d_3	0	0
d_6	d_4	d_3	0	0

1.adım:

İlk olarak tamamen farklı olan durumlardan başlanarak bunların kesiştiği yerler **X** olarak işaretlenir. Burada tamamen bağımsız olan durum d_2 ' dir. Bu nedenle tabloda d_2 olan satır ve sütun **X** olarak işaretlenir.



2. adım:

- Tablonun satır ve sütunlarının kesiştiği kutuya, durumlar farklıysa **X** aynıysa **✓** işareti konur.
- d_0 durumuyla en üstteki d_1 durumundan başlanarak farklı olan durumlar kutuya koşul olarak yazılarak aşağıya doğru devam edilir. d_N durumu değerlendirildikten sonra bir sonraki d_1 sütununa geçilerek bütün sütunlar bitene kadar işleme devam edilir. Böylece soldaki tablo elde edilir. Örneğin d_0 ile d_3 durumlarında $x=0$ girişi için d_0 , d_1 'e ve d_3 ise d_4 'e gittiği için farklıdır ve kutuya koşul olarak yazılır. $x=1$ girişi için d_0 , d_0 'a ve d_3 ise d_0 'a gittiği için aynıdır ve kutuya koşul olarak yazılmaz!

d_1	X					
d_2	X	X				
d_3	d_1, d_4	X	X			
d_4	X	✓	X	X		
d_5	d_1, d_4 d_0, d_3	X	X	d_0, d_3	X	
d_6	d_1, d_4 d_0, d_3	X	X	d_0, d_3	X	✓
	d_0	d_1	d_2	d_3	d_4	d_5

d_1	X					
d_2	X	X				
d_3	✓	X	X			
d_4	X	✓	X	X		
d_5	✓	X	X	✓	X	
d_6	✓	X	X	✓	X	✓
	d_0	d_1	d_2	d_3	d_4	d_5

Şimdiki durum	Gelecek durum		Çıkış	
	$x=0$	$x=1$	$x=0$	$x=1$
d_0	d_1	d_0	0	0
d_1	d_2	d_4	1	0
d_2	d_5	d_2	1	1
d_3	d_4	d_0	0	0
d_4	d_2	d_4	1	0
d_5	d_4	d_3	0	0
d_6	d_4	d_3	0	0

- Koşul yazılan kutular sorgulanarak eşdeğer olan ve olmayan durumlar belirlenir.
- Örneğin d_0-d_3 'teki " d_1, d_4 " koşulu için durum tablosuna bakılır. Her ikisinin de aynı girişler için aynı " d_2, d_4 " durumlar ve aynı çıkışlar olduğu görülür. d_1 ve d_4 durumları eşdeğerdir ve kesişim kutusuna ✓ yazılır.
- d_0-d_3 durumların eşdeğerliği d_1-d_4 durumlarının eşdeğerliğine koşullu olduğu için d_3-d_6 durumları da eşdeğerdir. Böylece bu durumların bulunduğu bütün kutular eşdeğer çıkar ve eşdeğerlik tablosu elde edilir.
- Sonuç olarak d_6, d_5 'e, d_5, d_3 'e, d_3, d_0 'a eşdeğerdir ve d_6, d_5, d_3 durumları yerine d_0 durumu kullanılır. Ayrıca d_4, d_1 'e eşdeğer olduğu için indirgenmiş durumlar d_0, d_1 ve d_2 olur.

$$d_0 \equiv d_3 \equiv d_5 \equiv d_6$$

$$d_1 \equiv d_4$$

d_1	X					
d_2	X	X				
d_3	d_1, d_4	X	X			
d_4	X	✓	X	X		
d_5	d_1, d_4 d_0, d_3	X	X	d_0, d_3	X	
d_6	d_1, d_4 d_0, d_3	X	X	d_0, d_3	X	✓
	d_0	d_1	d_2	d_3	d_4	d_5

d_1	X					
d_2	X	X				
d_3	✓	X	X			
d_4	X	✓	X	X		
d_5	✓	X	X	✓	X	
d_6	✓	X	X	✓	X	✓
	d_0	d_1	d_2	d_3	d_4	d_5

Şimdiki durum	Gelecek durum		Çıkış	
	$x=0$	$x=1$	$x=0$	$x=1$
d_0	d_1	d_0	0	0
d_1	d_2	d_4	1	0
d_2	d_5	d_2	1	1
d_3	d_4	d_0	0	0
d_4	d_2	d_4	1	0
d_5	d_4	d_3	0	0
d_6	d_4	d_3	0	0

İndirgenmemiş durum tablosu: (3 FF gerektirir)

Şimdiki durum	Gelecek durum		Çıkış	
	x=0	x=1	x=0	x=1
d_0	d_1	d_0	0	0
d_1	d_2	d_4	1	0
d_2	d_5	d_2	1	1
d_3	d_4	d_0	0	0
d_4	d_2	d_4	1	0
d_5	d_4	d_3	0	0
d_6	d_4	d_3	0	0

Soldaki tabloda
 $d_0=d_3=d_5=d_6$
ve $d_1=d_4$
alınır.



Şimdiki durum	Gelecek durum		Çıkış	
	x=0	x=1	x=0	x=1
d_0	d_1	d_0	0	0
d_1	d_2	d_1	1	0
d_2	d_0	d_2	1	1
d_0	d_1	d_0	0	0
d_1	d_2	d_1	1	0
d_0	d_1	d_0	0	0
d_0	d_1	d_0	0	0

İndirgenmiş durum tablosu: (2 FF gerektirir)

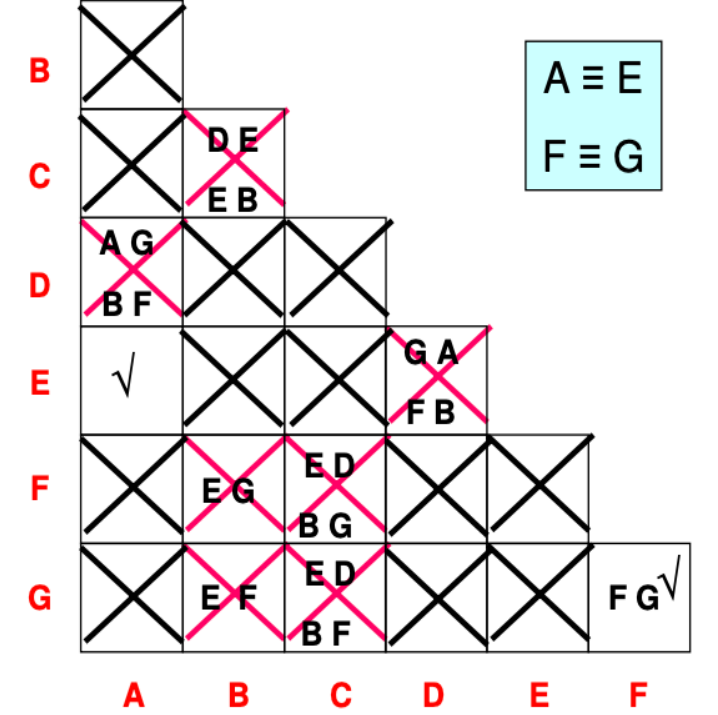
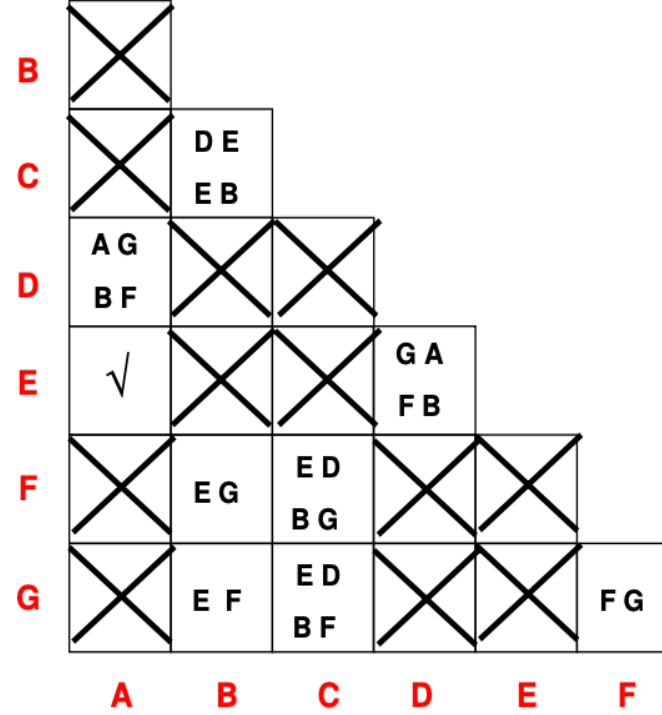
Şimdiki durum	Gelecek durum		Çıkış	
	x=0	x=1	x=0	x=1
d_0	d_1	d_0	0	0
d_1	d_2	d_1	1	0
d_2	d_0	d_2	1	1

Örnek : Aşağıda verilen durum tablosunu eşleştirme tablosu yöntemiyle indirgeyin.

Şimdiki durum	Gelecek durum		Çıkış	
	x=0	x=1	x=0	x=1
A	A	B	0	0
B	D	E	0	1
C	E	B	0	1
D	G	F	0	0
E	A	B	0	0
F	D	G	0	1
G	D	F	0	1

Durum indirgenmiş tablo:

Şimdiki durum	Gelecek durum		Çıkış	
	x=0	x=1	x=0	x=1
A	A	B	0	0
B	D	A	0	1
C	A	B	0	1
D	F	F	0	0
F	D	F	0	1

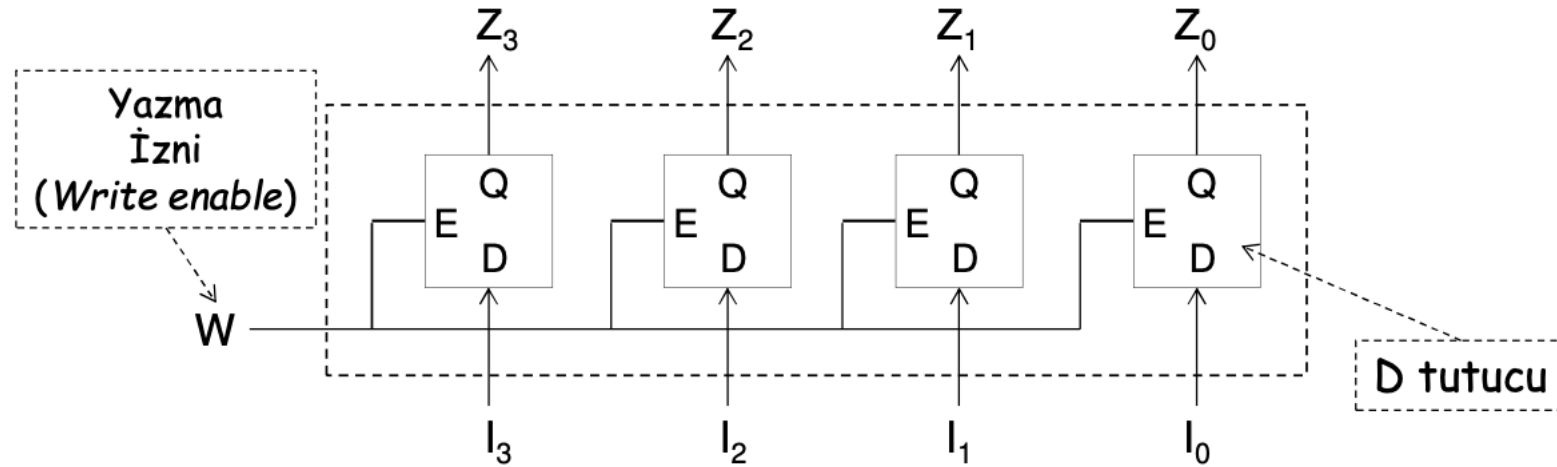


Saklayıcılar (Registers)

Saklayıcılar n bitlik ikili sayıları saklayabilen (tutabilen) bellek elemanlarıdır.

Örnek: 4 bit paralel saklayıcı

D tutucular (latch) kullanılmıştır.

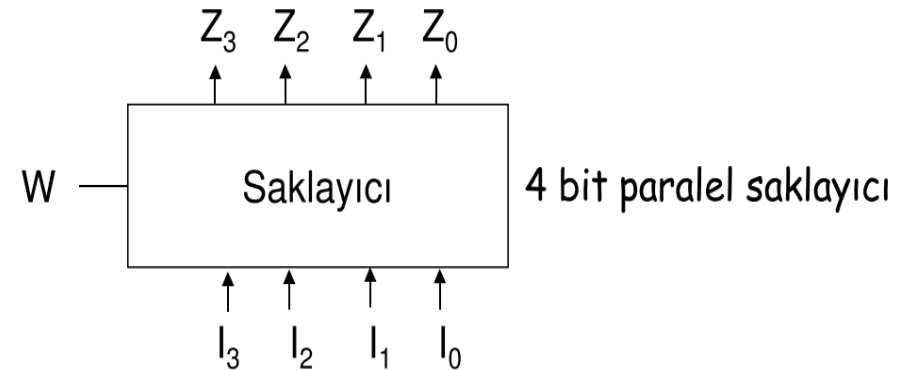


$W=0$ ise the saklayıcı o anki durumunu (içeriğini) korur.

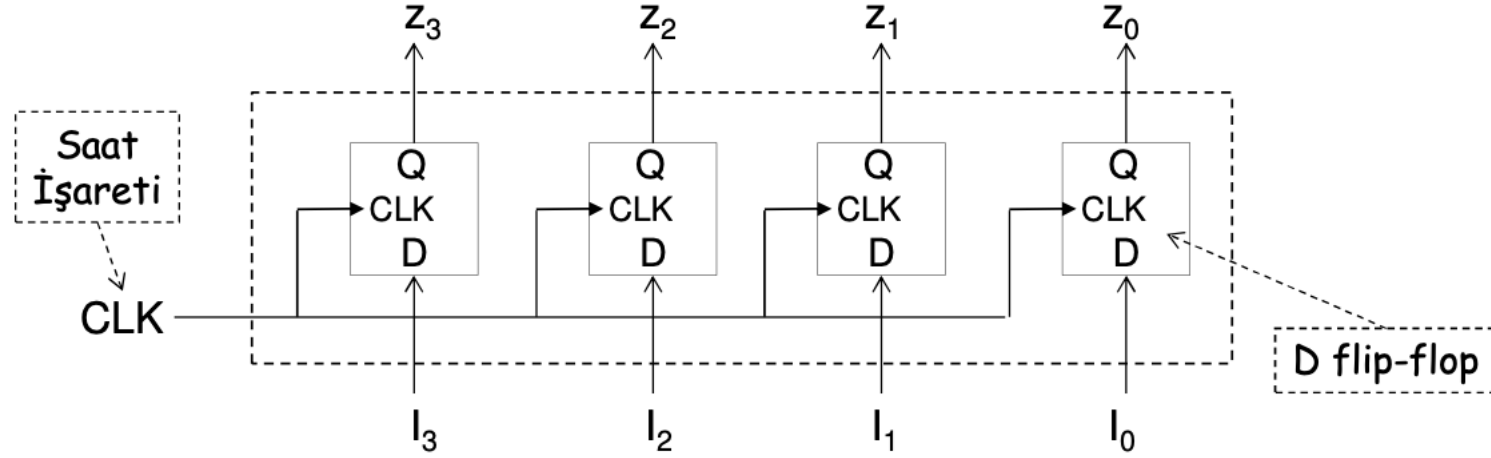
$W=1$ ise girişteki veri saklayıcıya yazılır.

$W = 0, Z(t+1) = Z(t)$

$W = 1, Z(t+1) = I(t)$

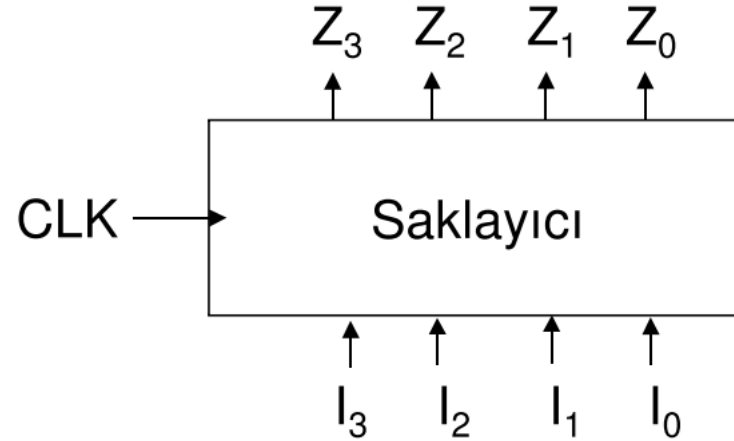


Örnek: Saat ile tetiklenen 4 bit paralel saklayıcı
D flip-flopları kullanılmıştır.



Saat işareti etkin olduğunda
girişteki veri saklayıcıya yazılır.

Saat ile tetiklenen 4bit paralel
saklayıcı:

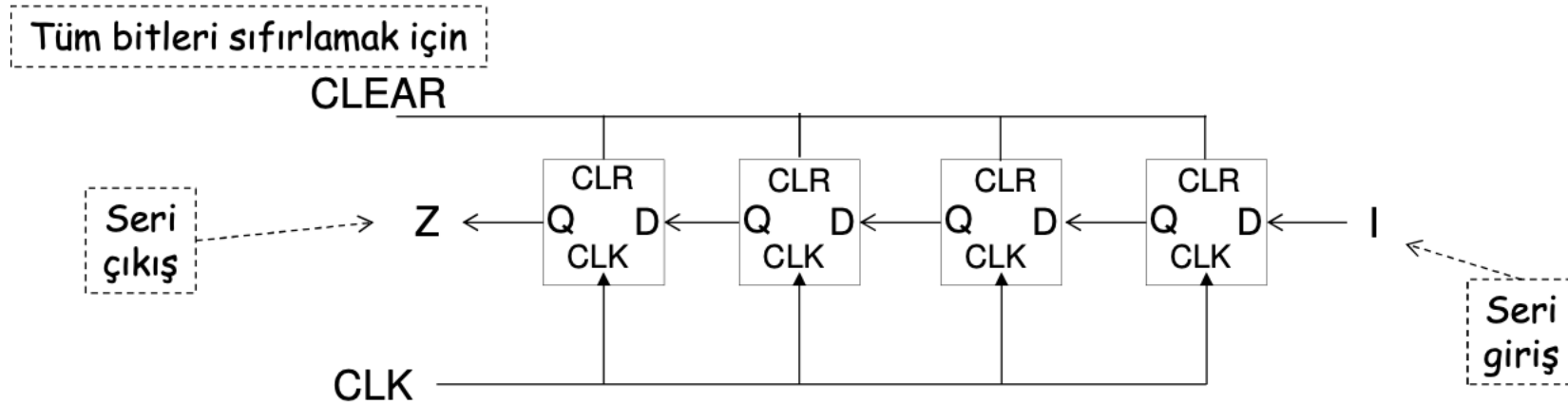


Not: Eğer tasarımda izin girişli (E) D flip-flopları kullanılırsa elde edilen saklayıcı hem saat işareti (CLK) hem de yazma izni (write enable -W) içerir.

Seri Ötelemeli Saklayıcılar (Serial Shift Registers)

Seri ötelemeli saklayıcılar ikili sayıları saklayabilirler ve saat işaretinin her etkin geçişinde bir bit sola veya sağa öteleyebilirler.

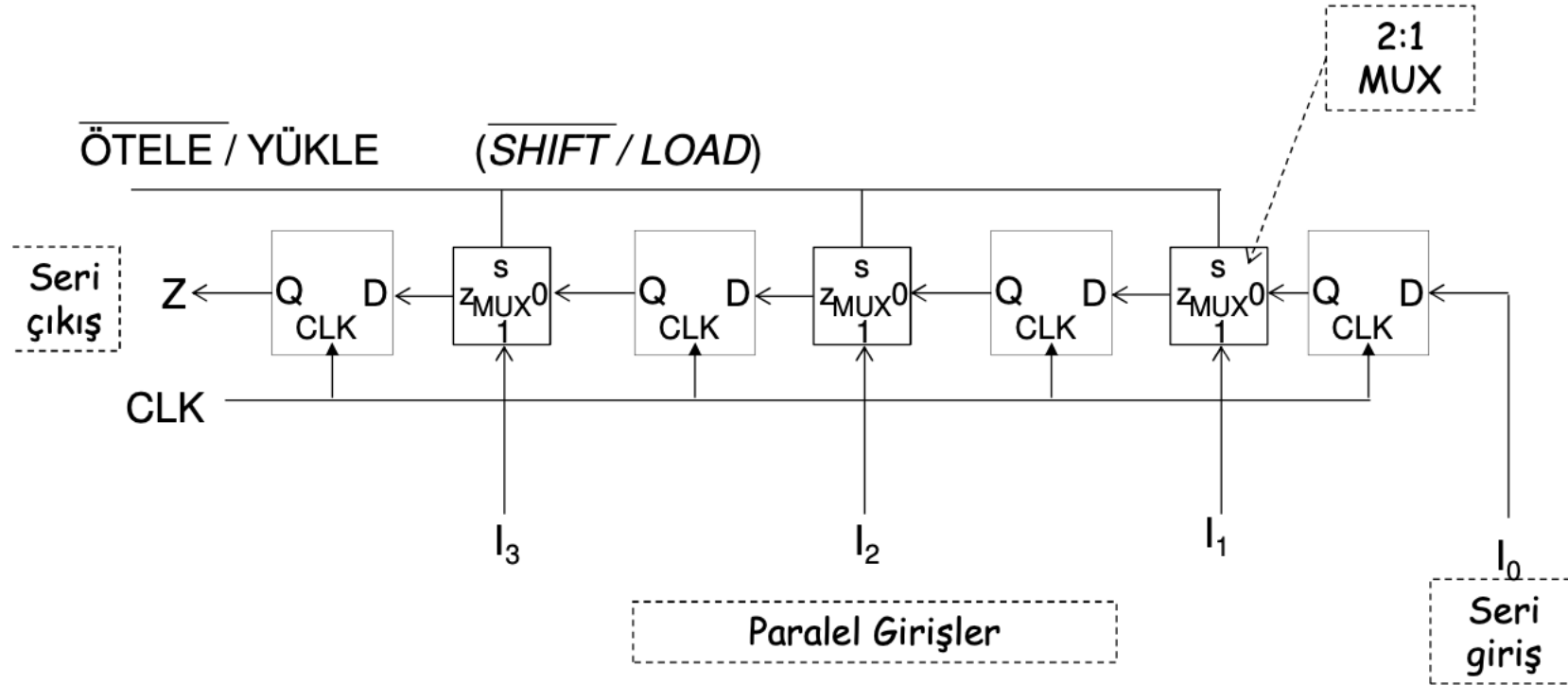
Örnek: 4-bit seri sola ötelemeli saklayıcı



Saat işaretinin her etkin geçişinde veri bir bit sola ötelenir. Flip-flopların bağlantıları (giriş/çıkış) yönleri değiştirilerek sağa ötelemeli saklayıcı gerçekleştirilebilir. Yukarıdaki tasarımda saklayıcıya başlangıç değeri (sıfırdan başka) yüklemek mümkün değildir.

Örnek: Paralel yüklenebilen 4-bit seri sola ötelemeli saklayıcı

Bu tasarımda saklayıcıya bir başlangıç değeri paralel olarak yüklenebilir.



$\overline{\text{SHIFT}} / \text{LOAD} = 0$: ötele

$\overline{\text{SHIFT}} / \text{LOAD} = 1$: yükle