

Sayısal Sistemler-H11CD1

Ardışık Lojik Devre Tasarım Örnekleri

Dr. Meriç Çetin
versiyon031220

Eş-zamanlamalı Saat Darbeli Ardışık Devreler

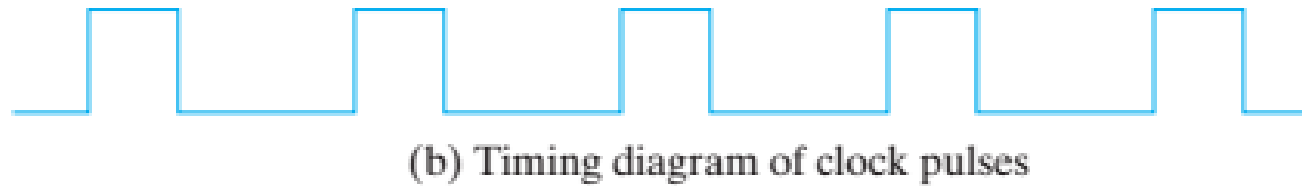
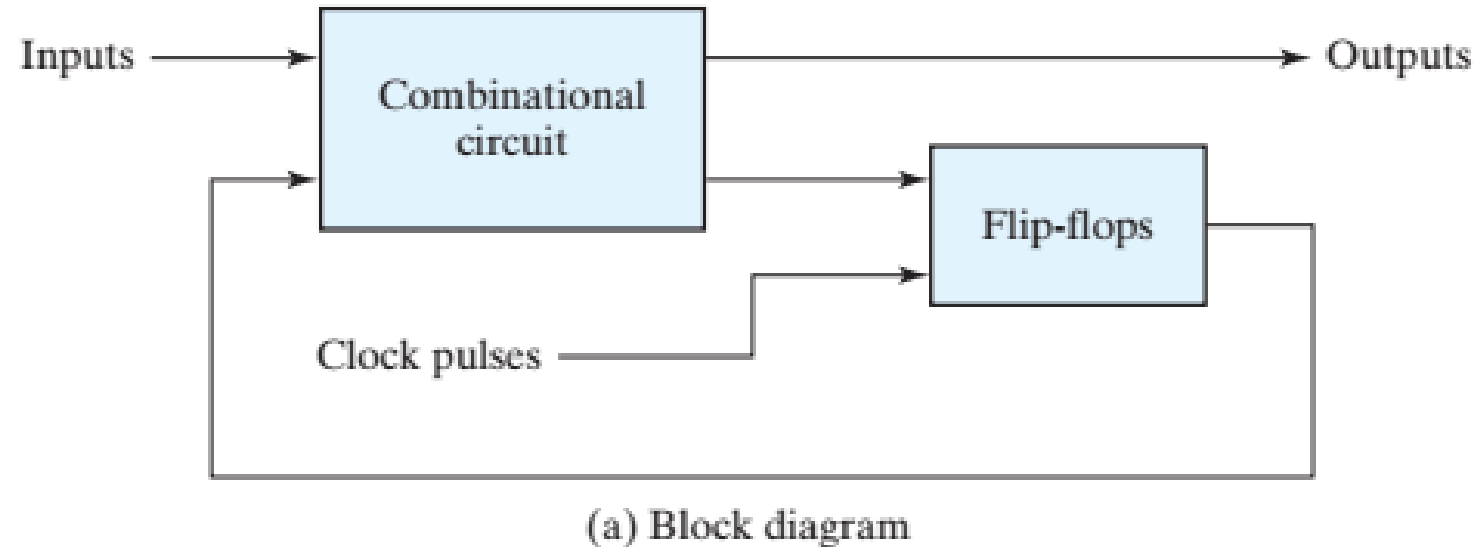


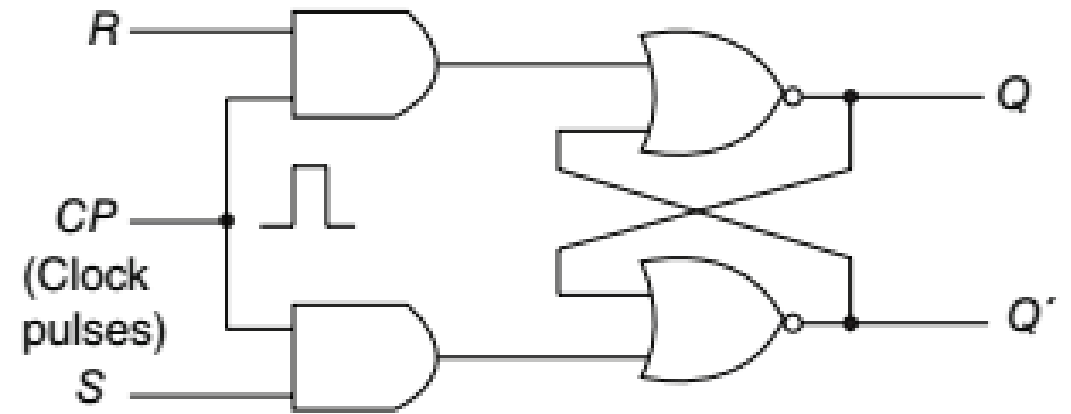
FIGURE 5.2
Synchronous clocked sequential circuit

Depolama Elemanları: Flip-Floplar

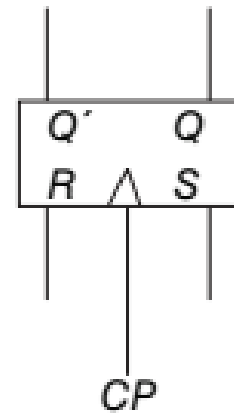
- Bir sayısal devredeki bir depolama elemanı, durumları değiştirmek için bir giriş sinyali tarafından yönlendirilene kadar ikili bir durumu süresiz olarak koruyabilir.
- Sinyal seviyeleriyle çalışan depolama elemanları saat geçişiyle kontrol edilen flip-floplar’dır.
- Temel flip-flop devreleri NOR veya NAND kapıları kullanılarak tasarlanır.

RS Flip-Flop

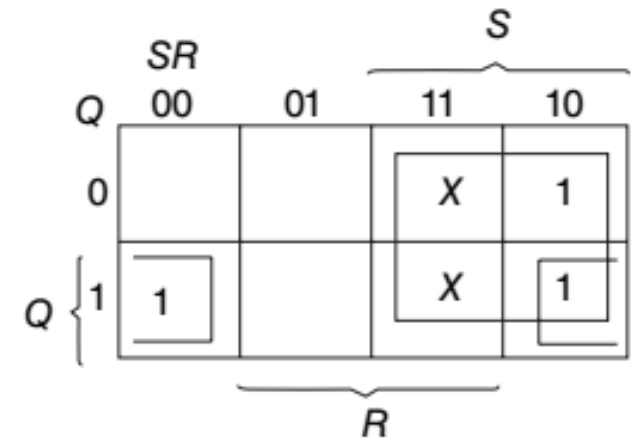
Önceki Durum				Sonraki Durum
CP	Q(t)	S	R	Q(t+1)
1	0	0	0	0 (Durum Değiştirmez)
1	0	0	1	0
1	0	1	0	1
1	0	1	1	Tanımsız
1	1	0	0	1 (Durum Değiştirmez)
1	1	0	1	0
1	1	1	0	1
1	1	1	1	Tanımsız



(a) Logic diagram



(b) Graphic symbol



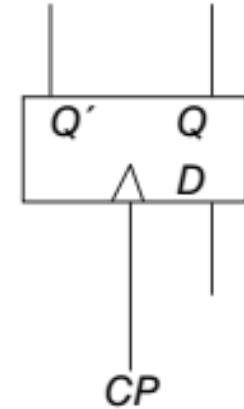
$$Q(r+1) = S + R'Q$$

$$SR = 0$$

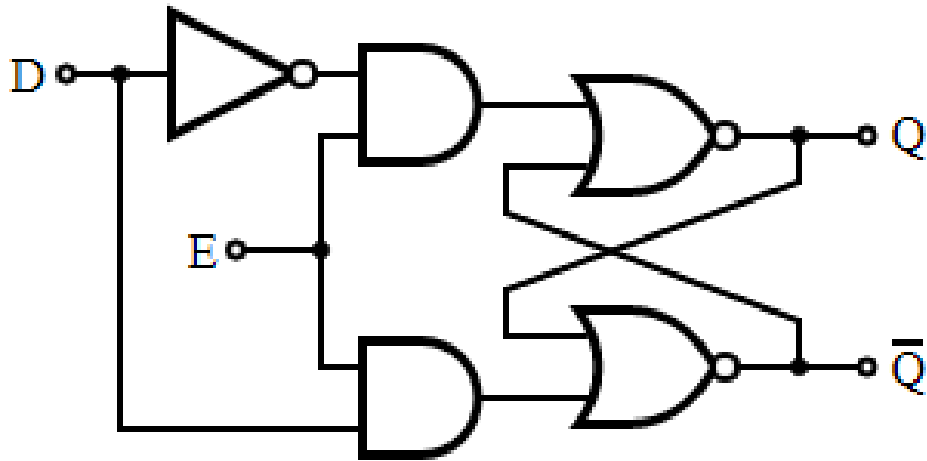
(d) Characteristic equation

D Tipi Flip-Flop

- D tipi flip-flop, RS tipi flip-flop'un girişlerinin değiştirilmesiyle elde edilir. D girişi doğrudan S girişine, R girişi ise S girişine bir invertor (tersleyici) ile bağlanarak D tipi flip-flop elde edilir.

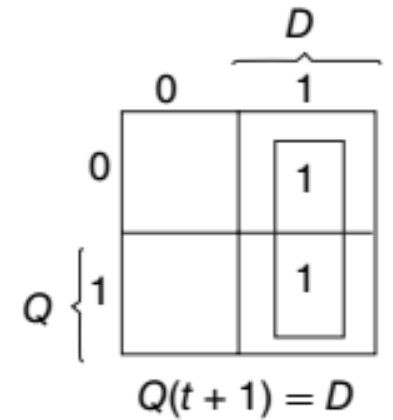


(b) Graphic symbol



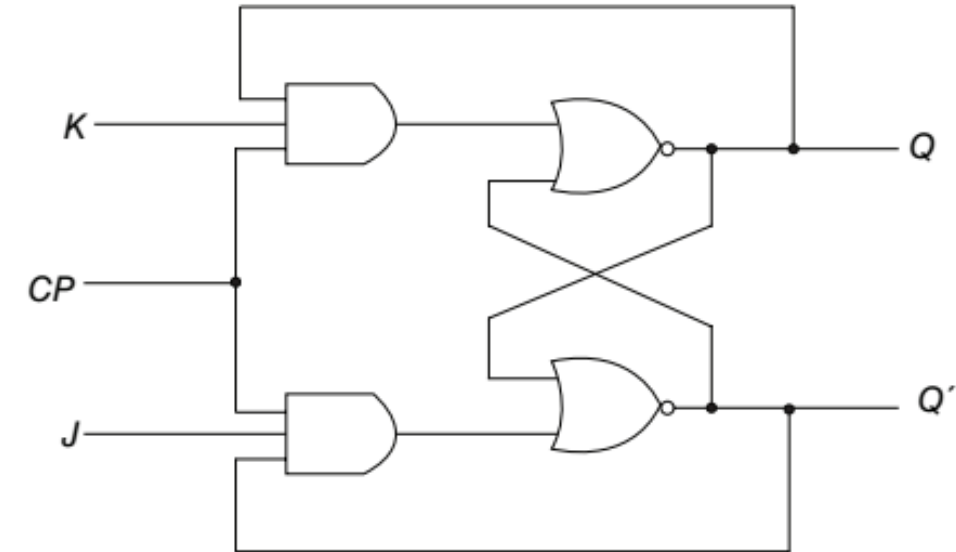
Q	D	$Q(t+1)$
0	0	0
0	1	1
1	0	0
1	1	1

(c) Characteristic table



(d) Characteristic equation

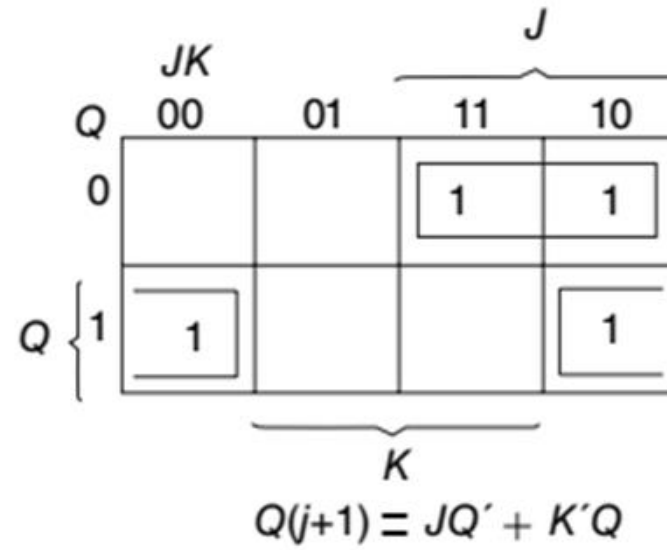
JK Tipi Flip-Flop



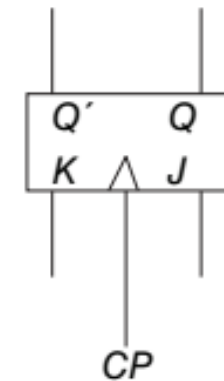
(a) Logic diagram

Q	J	K	$Q(t+1)$
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

(c) Characteristic table

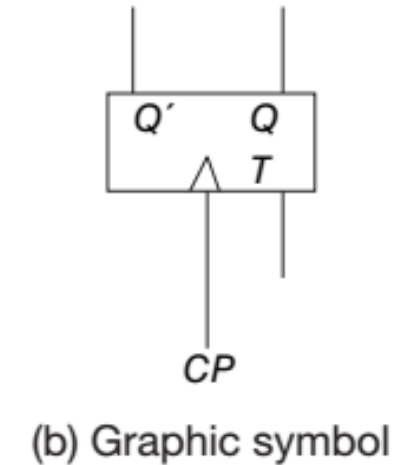
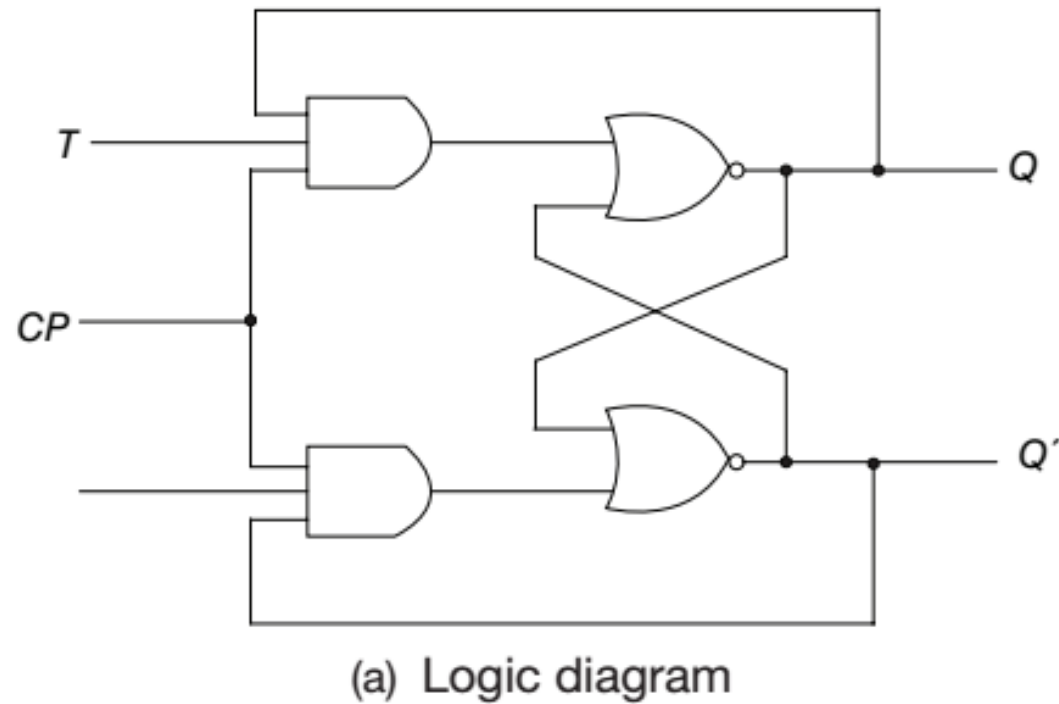


(d) Characteristic equation



(b) Graphic symbol

T (Trigger) Tipi Flip-Flop



Q	T	$Q(t+1)$
0	0	0
0	1	1
1	0	1
1	1	0

(c) Characteristic table

(d) Characteristic equation

			T
		0	1
Q	0		1
Q	1	1	

$$Q(t+1) = TQ' + T'Q$$

Kenar Tetiklemeli Flip-Flop'lar

- Flip-flop çıkışlarında herhangi bir hataya sebep vermemek için flip-flop'ların durum değiştirmeleri saat darbelerinin (clock pulse) uygulama zaman aralığı yerine saat darbelerinin
 - lojik «0» dan lojik «1» durumuna yükselme zamanlarında veya
 - lojik «1» den lojik «0» durumuna düşmeleri esnasında
- durum değiştirmelerini sağlamak amacıyla kenar tetiklemeli flip-flop'lar kullanılır.
- Bu sayede çıkış ile giriş arasındaki geri beslemeden dolayı meydana gelebilecek hatalı durum değiştirmeler ortadan kaldırılır.

Bir örnek

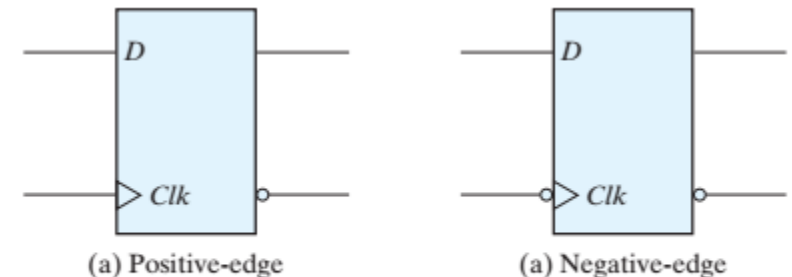
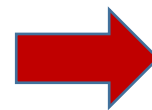


FIGURE 5.11

Graphic symbol for edge-triggered *D* flip-flop

Kenar Tetiklemeli Flip-Flop'lar

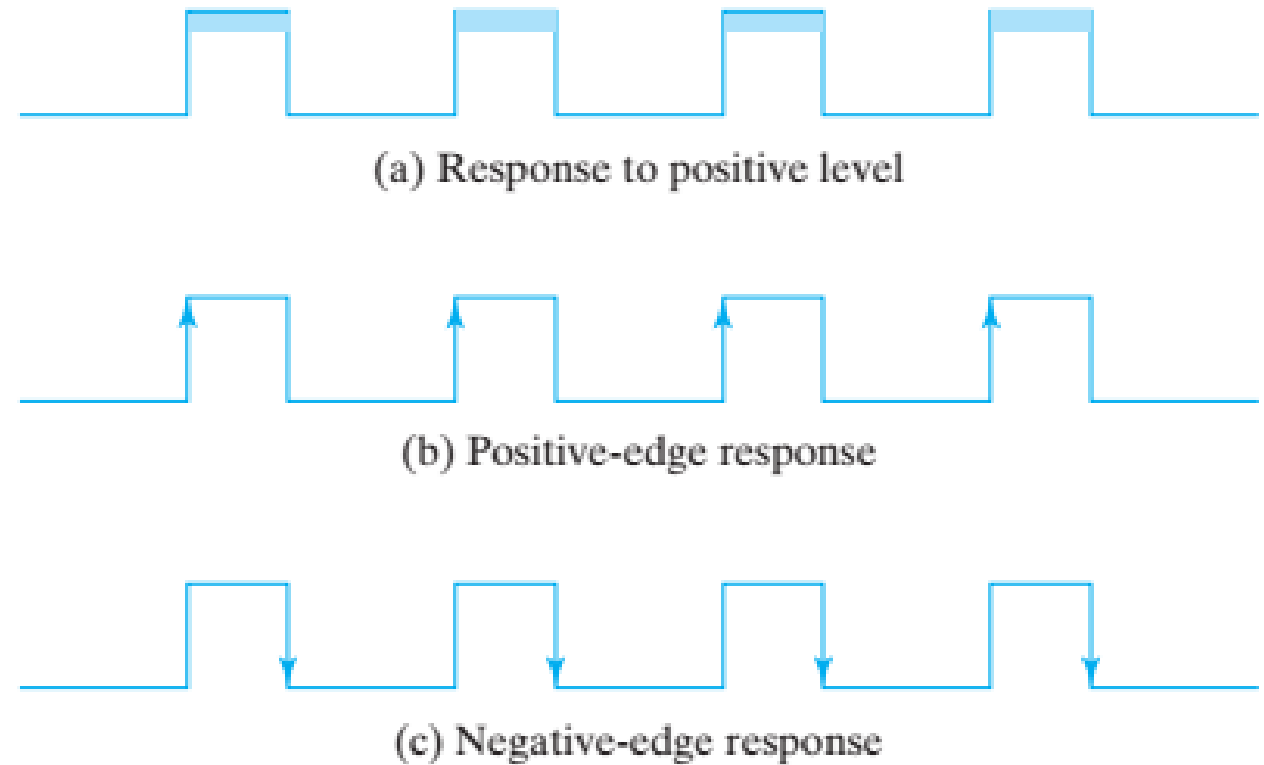


FIGURE 5.8

Clock response in latch and flip-flop

Flip-flop doğruluk tabloları

Flip-flop characteristic tables

S	R	$Q(t+1)$
0	0	$Q(t)$
0	1	0
1	0	1
1	1	?

(a) RS

J	K	$Q(t+1)$
0	0	$Q(t)$
0	1	0
1	0	1
1	1	$Q'(t)$

(b) JK

D	$Q(t+1)$
0	0
1	1

(c) D

T	$Q(t+1)$
0	$Q(t)$
1	$Q'(t)$

(d) T

Flip-flop durum geiş tabloları

Flip-flop excitation tables

$Q(t)$	$Q(t+1)$	S	R
0	0	0	X
0	1	1	0
1	0	0	1
1	1	X	0

(a) RS

$Q(t)$	$Q(t+1)$	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

(b) JK

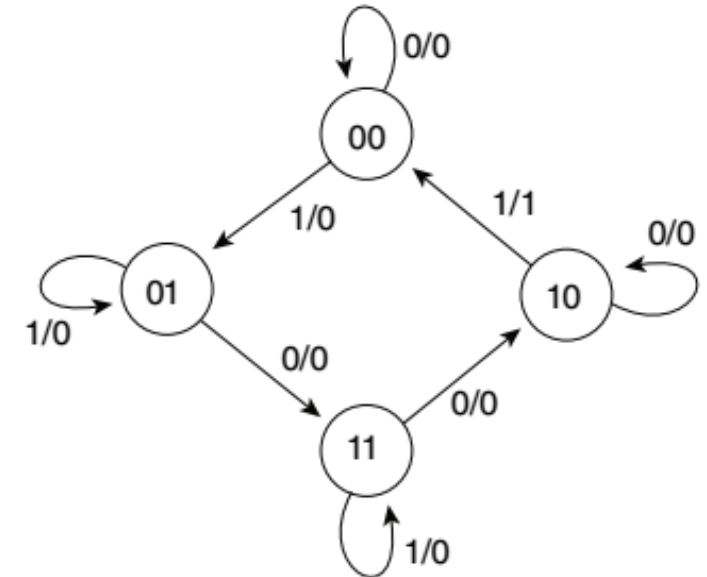
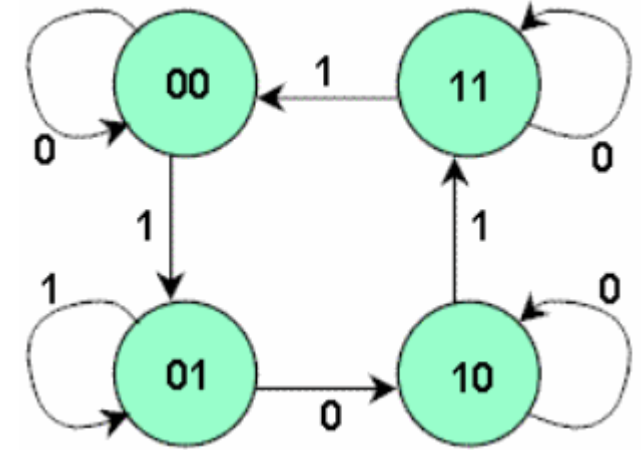
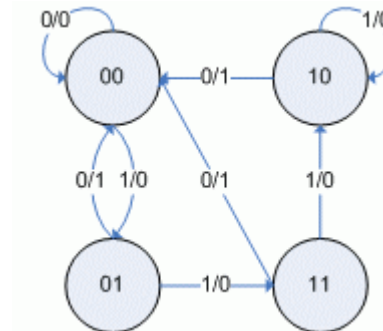
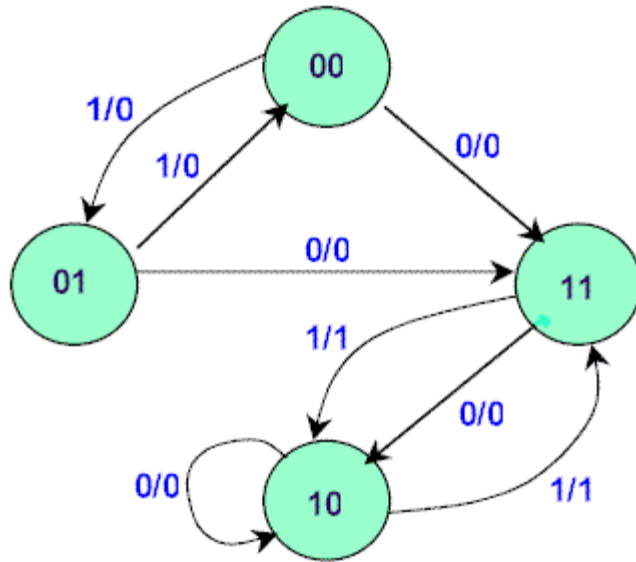
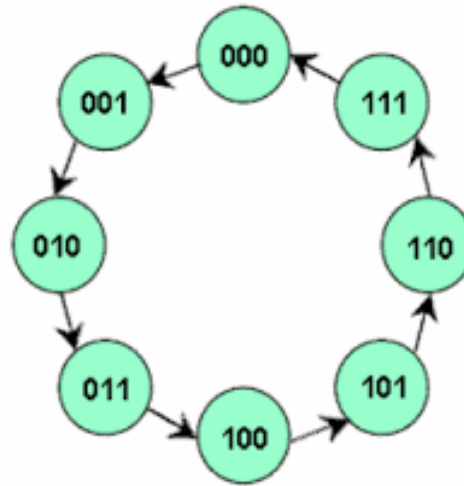
$Q(t)$	$Q(t+1)$	D
0	0	0
0	1	1
1	0	0
1	1	1

(c) D

$Q(t)$	$Q(t+1)$	T
0	0	0
0	1	1
1	0	1
1	1	0

(d) T

Durum diyagramlarına örnekler



Durum tablosundan durum diyagramına geçiş

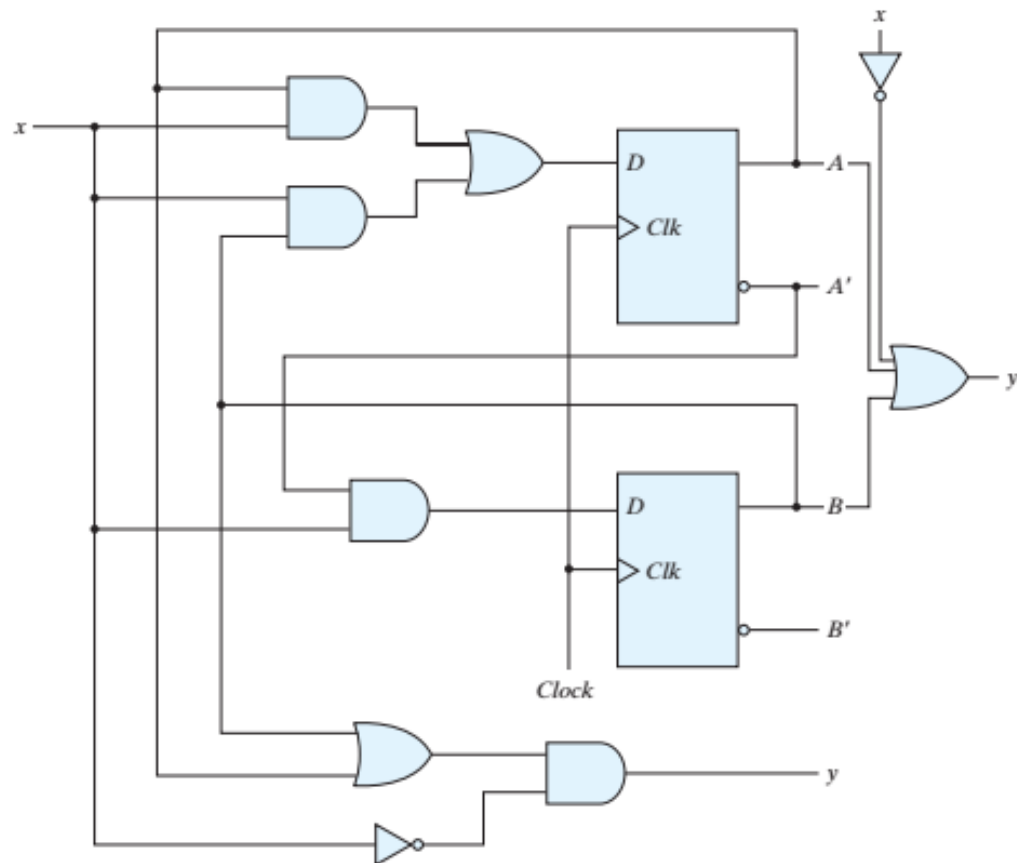


FIGURE 5.15
Example of sequential circuit

Table 5.3

Second Form of the State Table

Present State		Next State				Output	
		$x = 0$		$x = 1$		$x = 0$	$x = 1$
A	B	A	B	A	B	y	y
0	0	0	0	0	1	0	0
0	1	0	0	1	1	1	0
1	0	0	0	1	0	1	0
1	1	0	0	1	0	1	0

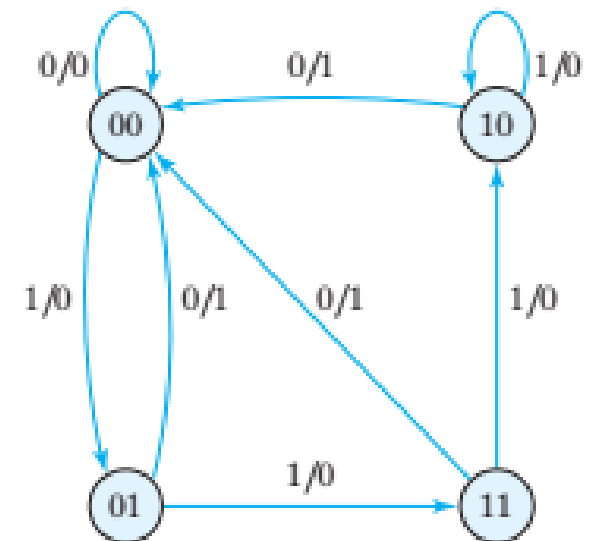


FIGURE 5.16
State diagram of the circuit of Fig. 5.15

Durum İndirgeme

- Ardışık bir devrede flip-flop sayısındaki azalma, durum indirgeme problemi olarak adlandırılır.
- Durum indirgeme algoritmaları, harici giriş-çıkış gereksinimlerini değiştirmeden tutarken, bir durum tablosundaki durumların sayısını azaltmak için prosedürlerle ilgilidir.
- m adet flip-flop 2^m adet durum ürettiğinden, durum sayısındaki bir azalma, flip-flopların sayısında bir azalmaya neden olabilir (veya olmayabilir).
- Flip-flop sayısını azaltmada öngörülemez bir etki, bazen eşdeğer devrenin (daha az flip-flop ile) bir sonraki durumunu ve çıkış mantığını gerçekleştirmek için daha fazla kombinasyon kapısı gerektirmesidir.

Durum İndirgeme-örnek

Table 5.7
Reducing the State Table

Present State	Next State		Output	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
<i>a</i>	<i>a</i>	<i>b</i>	0	0
<i>b</i>	<i>c</i>	<i>d</i>	0	0
<i>c</i>	<i>a</i>	<i>d</i>	0	0
<i>d</i>	<i>e</i>	<i>f</i>	0	1
<i>e</i>	<i>a</i>	<i>f</i>	0	1
<i>f</i>	<i>e</i>	<i>f</i>	0	1

e,g →

Table 5.8
Reduced State Table

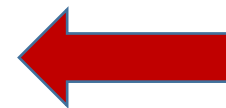
Present State	Next State		Output	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
<i>a</i>	<i>a</i>	<i>b</i>	0	0
<i>b</i>	<i>c</i>	<i>d</i>	0	0
<i>c</i>	<i>a</i>	<i>d</i>	0	0
<i>d</i>	<i>e</i>	<i>d</i>	0	1
<i>e</i>	<i>a</i>	<i>d</i>	0	1

d,f →

- Durum tablosunu inceleyerek, aynı sonraki duruma giden ve her iki giriş kombinasyonu için aynı çıktıya sahip iki mevcut durumu ararız.

Table 5.6
State Table

Present State	Next State		Output	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
<i>a</i>	<i>a</i>	<i>b</i>	0	0
<i>b</i>	<i>c</i>	<i>d</i>	0	0
<i>c</i>	<i>a</i>	<i>d</i>	0	0
<i>d</i>	<i>e</i>	<i>f</i>	0	1
<i>e</i>	<i>a</i>	<i>f</i>	0	1
<i>f</i>	<i>g</i>	<i>f</i>	0	1
<i>g</i>	<i>a</i>	<i>f</i>	0	1

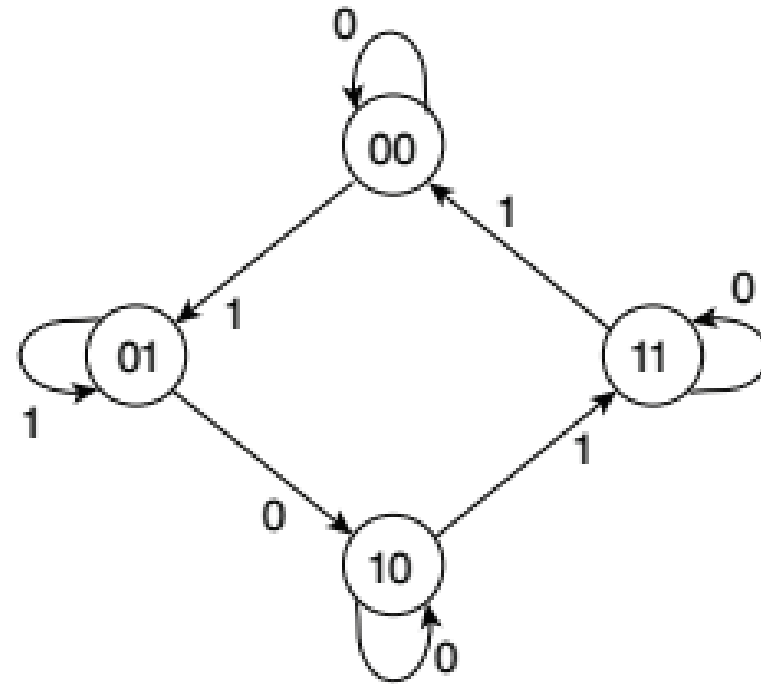


Ardışık Lojik Devre Tasarım Prosedürü

- Ardışık lojik devre tasarımı için şu yol takip edilmelidir:
- Devre davranışı tanımlanır. Bu, durum diyagramlarıyla belirlenir.
- Elde edilen değerler durum tablosuna taşınır.
- Gerekli flip-flop sayısı ve flip-flop türü belirlenir.
- Karnaugh veya diğer indirgeme metotları kullanılarak kombinasyonel devre çıkış ve flip-flop giriş denklemleri elde edilir.
- Elde edilen bu sonuçlara göre lojik devre tasarımı yapılır.

Ardışık lojik devre tasarım örneği

- Aşağıda durum diyagramı verilen lojik devreyi J-K flip-floplarını kullanarak gerçekleştiriniz.

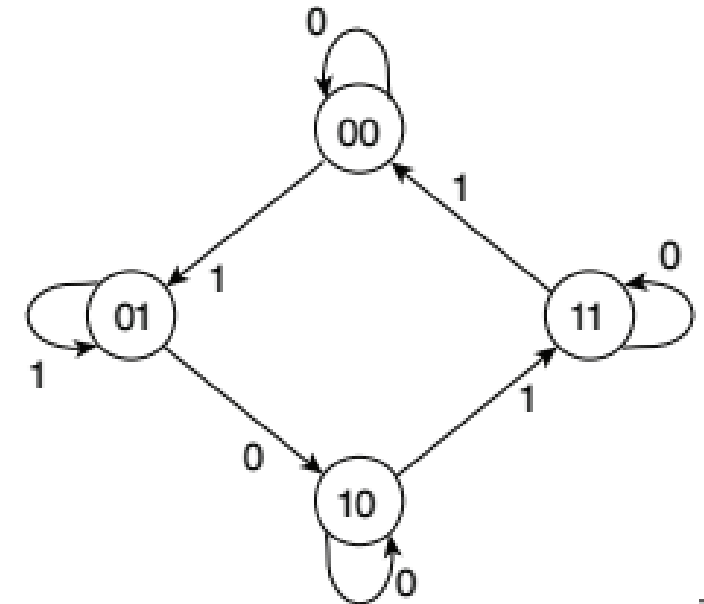


State diagram

Ardışık lojik devre tasarım örneği

- Durum diyagramından durum tablosuna geçiş:

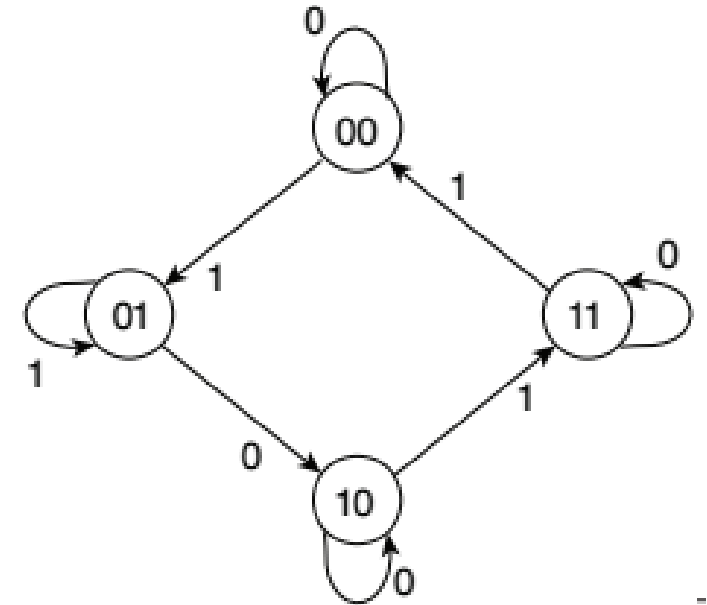
Durum Tablosu					
Önceki Durum		Sonraki Durum			
A	B	x=0		x=1	
		A	B	A	B
0	0				
0	1				
1	0				
1	1				



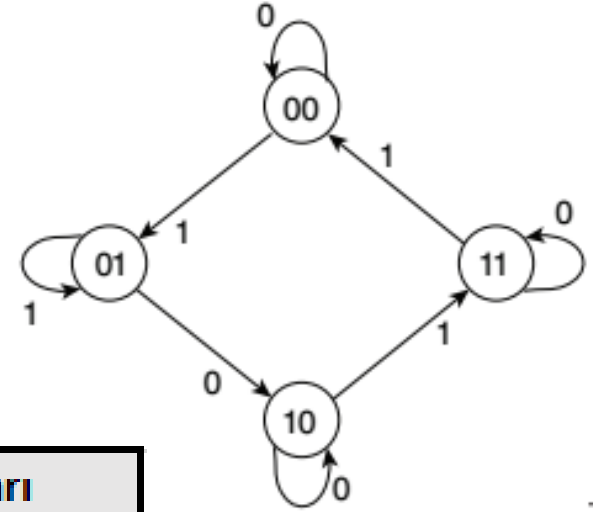
Ardışık lojik devre tasarım örneği

- Durum diyagramından durum tablosuna geçiş:

Durum Tablosu					
Önceki Durum		Sonraki Durum			
A	B	x=0		x=1	
		A	B	A	B
0	0	0	0	0	1
0	1	1	0	0	1
1	0	1	0	1	1
1	1	1	1	0	0

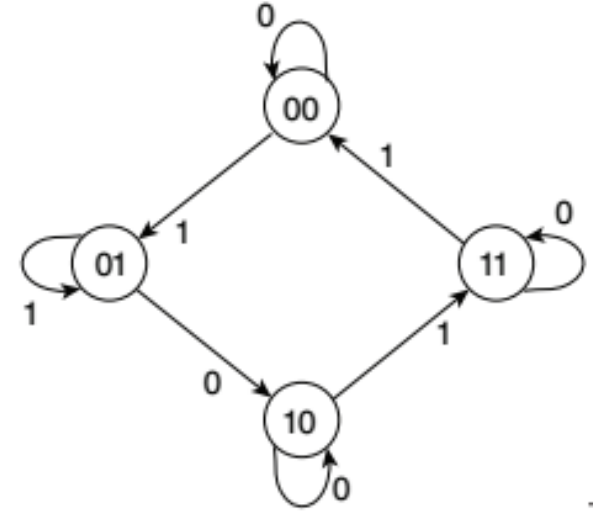


Ardışık lojik devre tasarım örneği



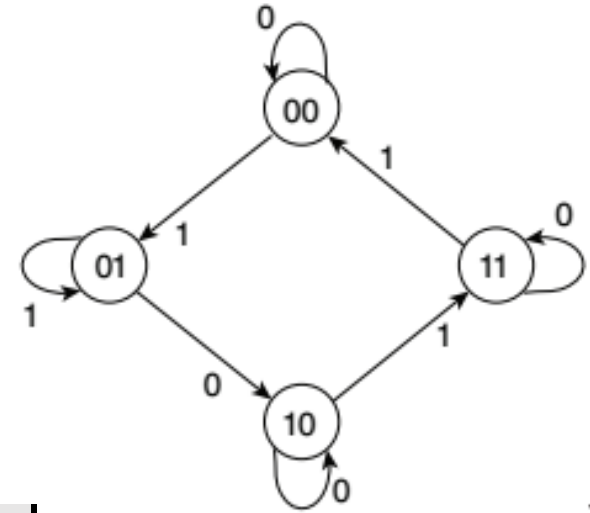
Kombinasyonel Devre Girişleri			Sonraki Durum		Kombinasyonel Devre Çıktıları			
Önceki Durum		Giriş			Flip-flop Girişleri			
A	B	x	A	B	JA	KA	JB	KB
0	0	0						
0	0	1						
0	1	0						
0	1	1						
1	0	0						
1	0	1						
1	1	0						
1	1	1						

Ardışık lojik devre tasarım örneği



Kombinasyonel Devre Girişleri			Sonraki Durum		Kombinasyonel Devre Çıktıları			
Önceki Durum		Giriş			Flip-flop Girişleri			
A	B	x	A	B	JA	KA	JB	KB
0	0	0	0	0				
0	0	1	0	1				
0	1	0	1	0				
0	1	1	0	1				
1	0	0	1	0				
1	0	1	1	1				
1	1	0	1	1				
1	1	1	0	0				

Ardışık lojik devre tasarım örneği

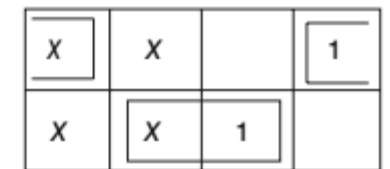
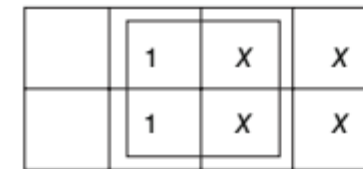
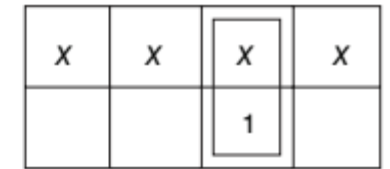
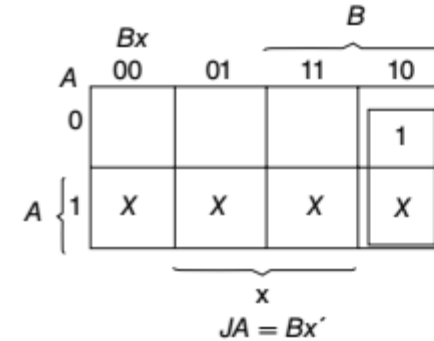


Kombinasyonel Devre Girişleri			Sonraki Durum		Kombinasyonel Devre Çıktıları			
Önceki Durum		Giriş			Flip-flop Girişleri			
A	B	x	A	B	JA	KA	JB	KB
0	0	0	0	0	0	X	0	X
0	0	1	0	1	0	X	1	X
0	1	0	1	0	1	X	X	1
0	1	1	0	1	0	X	X	0
1	0	0	1	0	X	0	0	X
1	0	1	1	1	X	0	1	X
1	1	0	1	1	X	0	X	0
1	1	1	0	0	X	1	X	1

Ardışık lojik devre tasarım örneği

- Flip-flop girişlerini elde etmek için Karnaugh diyagramlarından faydalanalım.

Kombinasyonel Devre Girişleri			Sonraki Durum		Kombinasyonel Devre Çıktıları			
Önceki Durum		Giriş			Flip-flop Girişleri			
A	B	x	A	B	JA	KA	JB	KB
0	0	0	0	0	0	X	0	X
0	0	1	0	1	0	X	1	X
0	1	0	1	0	1	X	X	1
0	1	1	0	1	0	X	X	0
1	0	0	1	0	X	0	0	X
1	0	1	1	1	X	0	1	X
1	1	0	1	1	X	0	X	0
1	1	1	0	0	X	1	X	1



Ardışık lojik devre tasarım örneği

- Ardışık lojik devreyi çizelim:

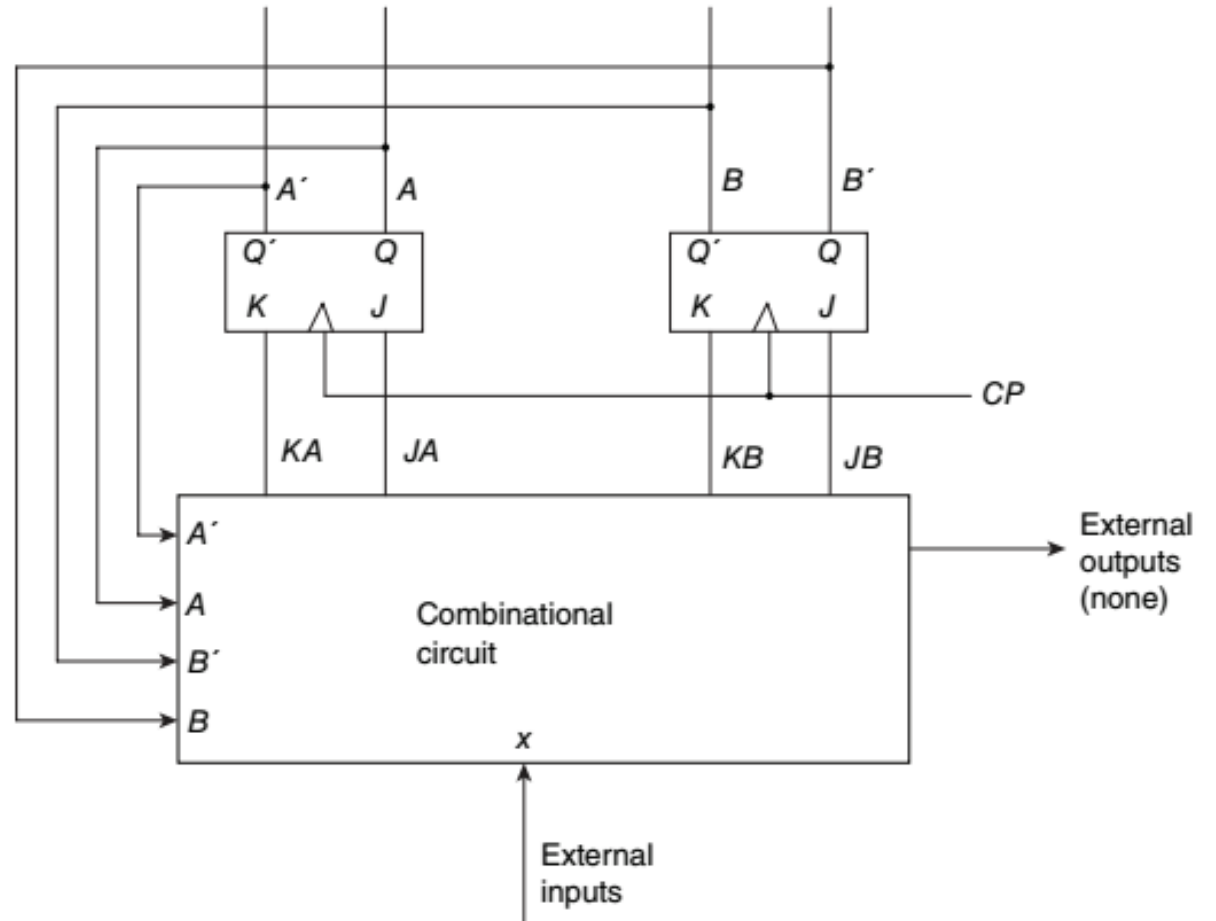
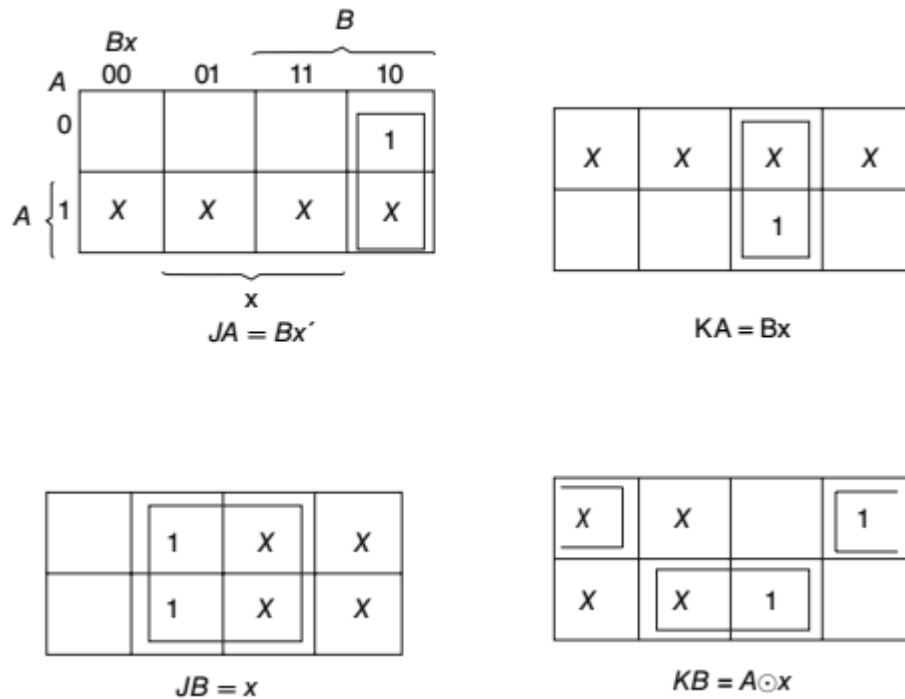


Figure 6-22 Block diagram of sequential circuit

Ardışık lojik devre tasarım örneği

- Ardışık lojik devreyi çizelim:

		Bx			
		B			
A		00	01	11	10
	0				1
	1	X	X	X	X

$JA = Bx'$

X	X	X	X
		1	

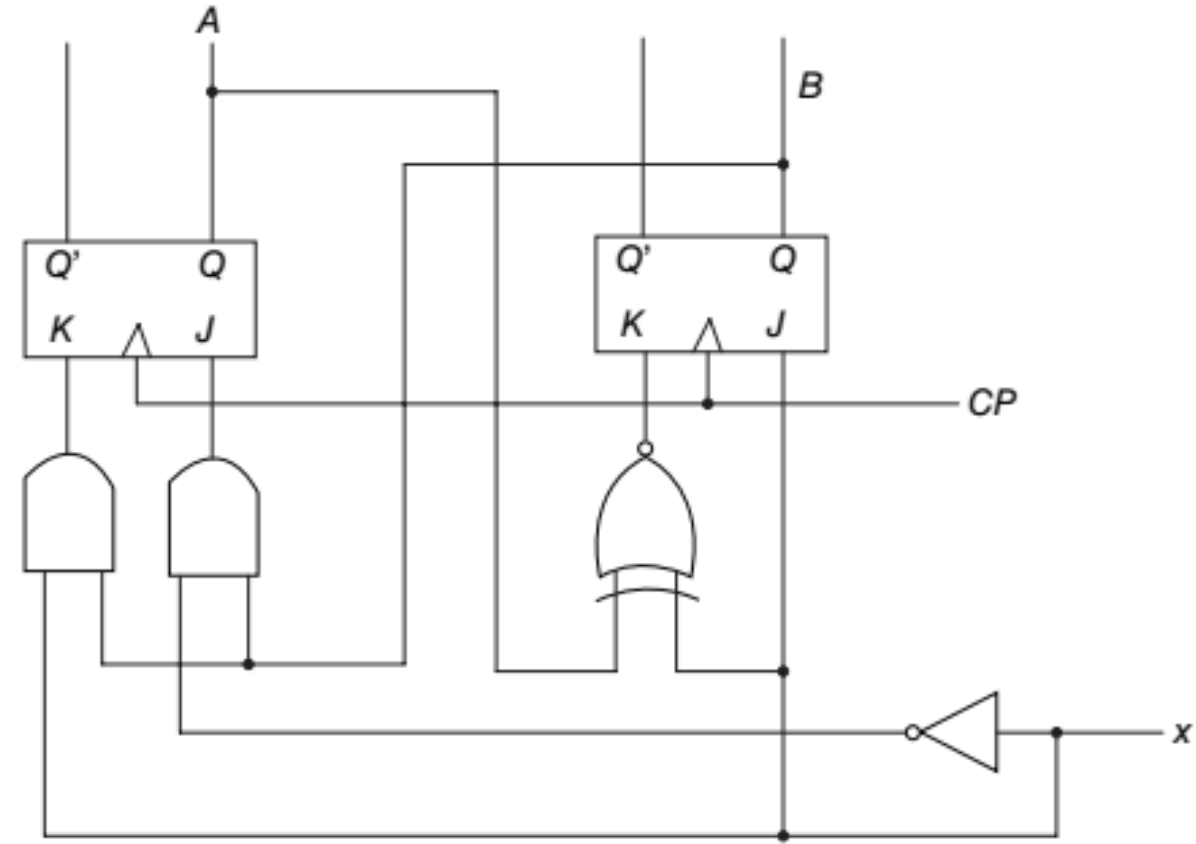
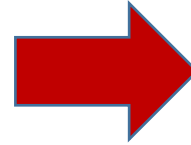
$KA = Bx$

	1	X	X
	1	X	X

$JB = x$

X	X		1
X	X	1	

$KB = A \odot x$



Logic diagram of sequential circuit