

Ardışıl Devre Sentezi (Sequential Circuit Design)

Ardışıl devre tasarımı prosedürü:

Adım 1:

Problemin tanımına uygun olarak durum tablosunu yapılır. Tablo şimdiki durumları, girişleri, gelecek durumları ve çıkışları içermeli. (Bazen öncelikle diyagramı çizip ardından tabloya dönüştürmek daha kolay olmaktadır.)

Adım 2:

Durum tablosundaki durumlara binary kod atanır. n tane durum varsa, binary kodlar en az $\lceil \log_2 n \rceil$ digit olmalıdır ve devrede en az $\lceil \log_2 n \rceil$ tane flip-flop olacaktır.

Adım 3:

Her bir flip-flop için giriş değerleri bulunur. Bunun için flip-flop ters karakteristik tabloları kullanılabilir.

Adım 4:

Flip-flop giriş ve çıkışları için basitleştirilmiş ifadeler bulunur.

Adım 5:

Devre kurulur.

Dizi Yakalayıcısı (Sequence recognizers)

- Bir **dizi yakalayıcısı** devrenin girişinde özel bir bit dizisini arayan özel bir ardışıl devredir.
- Bu örnek yakalayıcı devrenin tek bir girişi vardır: X
Girişin her bir biti bir saat periyotunda sağlanır.
- Tek bir çıkış vardır: Z. İstenen bit dizisi bulunduğunda 1 çıkışını verir.
- **Örnek:** Aranacak olan bit dizisi: 1001

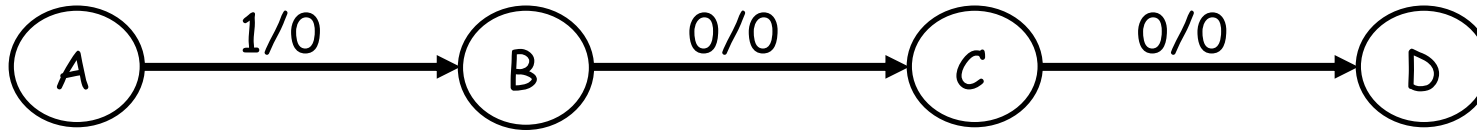
Girişler: 1 1 **1 0 0 1** 1 0 **1 0 0 1** 0 0 1 1 0 ...

Çıkışlar: 0 0 0 0 0 **1** 0 0 0 0 0 **1** 0 0 **1** 0 0 ...

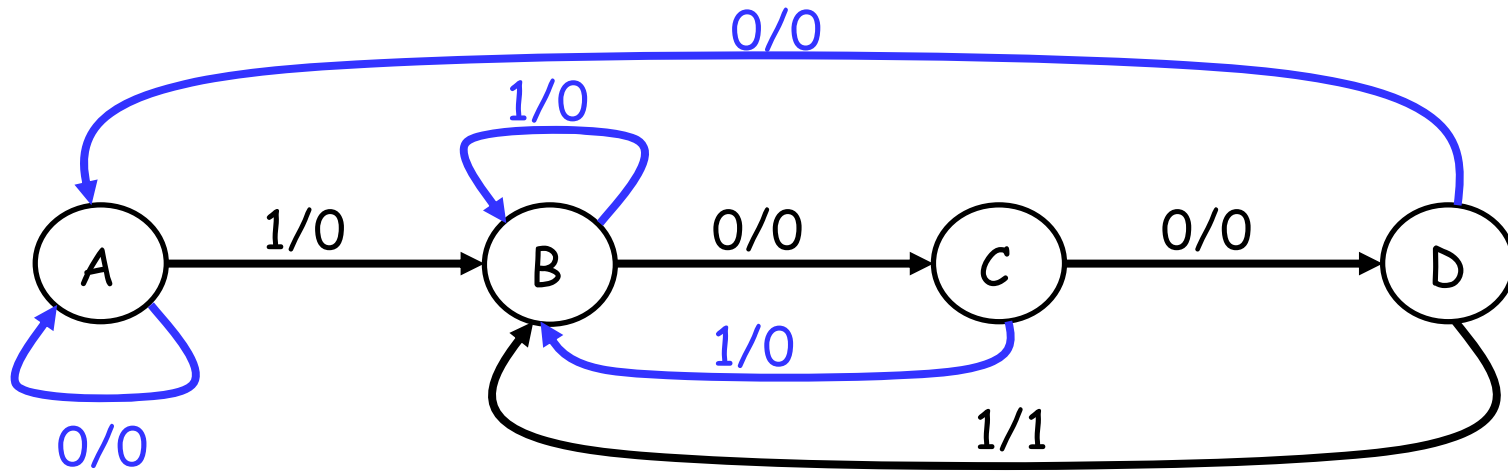
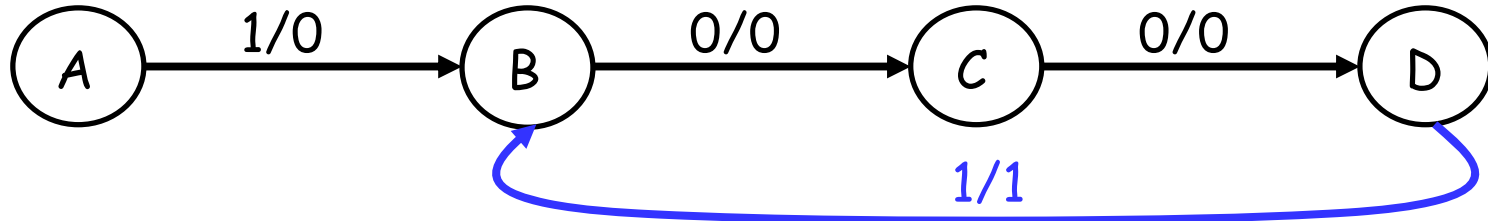
- Bu tasarımda bir ardışıl devre tasarlanması gerekmektedir. Çünkü devrenin diziyi yakalayabilmesi için birkaç önceki çıkışı da hatırlayabilmesi gerekmektedir.

Adım 1: Durum tablosu yapma

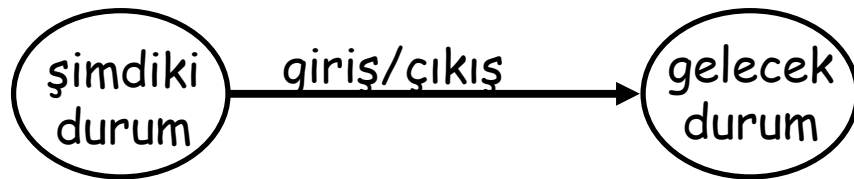
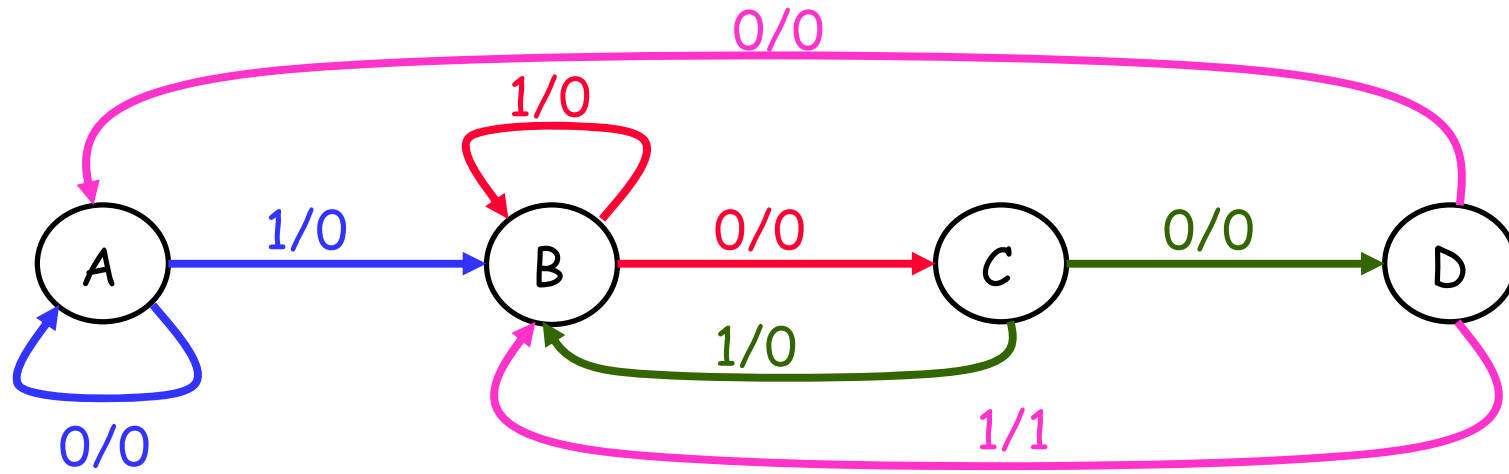
- Dizi yakalayıcısı için öncelikle durum diyagramı yapıp, ardından tabloyu oluşturalım:
 - Önceki saat periyotlarındaki girişleri hatırlamalı
 - Örneğin, eğer önceki üç giriş 100 ve şu anki giriş 1 ise, çıkış 1 olacaktır.
 - Genellikle, bu dizinin belli kısımlarının oluşumunu incelemek gerekiyor—bu örnekte: 1, 10, and 100
- Durum diyagramını çizmeye başlayalım:



Durum	Anlamı
A	İstenilen dizinin (1001) başlangıcı henüz görülmedi.
B	İstenilen dizinin ilk biti (1) görüldü.
C	İstenilen dizinin ilk iki biti (10) görüldü.
D	İstenilen dizinin ilk üç biti (100) görüldü.



Not: Her bir nod için *iki* çıkış oku vardır: $X=0$ ve $X=1$ girişi için.

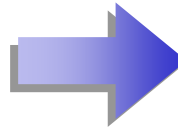


Şimdiki durum	Giriş	Gel. durum	Çıkış
A	0	A	0
A	1	B	0
B	0	C	0
B	1	B	0
C	0	D	0
C	1	B	0
D	0	A	0
D	1	B	1

Adım 2: Durumlara binary kod atamak

- Dört durumumuz var: ABCD. O halde en az iki flip-flop'a ihtiyacımız var: Q_1Q_0
- Bunun için en kolay yöntem: A için $Q_1Q_0 = 00$, B için 01, C için 10 ve D için 11.

Şimdiki Durum	Giriş	Gelecek Durum	Çıkış
A	0	A	0
A	1	B	0
B	0	C	0
B	1	B	0
C	0	D	0
C	1	B	0
D	0	A	0
D	1	B	1



Şimdiki Durum		Giriş X	Gelecek Durum		Çıkış Z
Q_1	Q_0		Q_1	Q_0	
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	1	0	0
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	0	1	0
1	1	0	0	0	0
1	1	1	0	1	1

Adım 3: Flip-flop giriş değerlerinin bulunması

- Bu hangi tip flip-flop kullandığınıza göre değişir!
- İki tane JK FF kullanacağız. Her bir flip-flop için, şimdiki ve gelecek durumlarına bakarak bu durum değişikliğini sağlayacak olan J_i ve K_i girişlerinin ne olması gerektiği belirlenir.

Şimdiki Durum		Giriş X	Gelecek Durum		Flip flop girişleri				Çıkış Z
Q_1	Q_0		Q_1	Q_0	J_1	K_1	J_0	K_0	
0	0	0	0	0					0
0	0	1	0	1					0
0	1	0	1	0					0
0	1	1	0	1					0
1	0	0	1	1					0
1	0	1	0	1					0
1	1	0	0	0					0
1	1	1	0	1					1

Flip-flop giriş değerlerinin bulunması

- JK FF için, giriş değerlerinin bulunması biraz "tricky". JK FF'a ait karakteristik tabloyu hatırlayalım:

J	K	$Q(t+1)$	İşlem
0	0	$Q(t)$	Değişmez
0	1	0	Reset
1	0	1	Set
1	1	$Q'(t)$	Tümlleme

- Eğer JK flip-flop'un şimdiki durumu 0 ise ve biz gelecek durumu 1 yapmak istiyorsak, bu durumda JK girişleri için *iki* seçenek söz konusudur:
 - JK= 10, gelecek durumu doğrudan 1 yapar.
 - JK=11, şimdiki durumun (0) tümleyenini alarak 1 gelecek durumu 1 yapar.
- O halde 0'dan 1'e değişiklik için, J=1 yapılmalı, ancak K 0 veya 1 olabilir.
- Benzer şekilde, diğer olası durum geçişleri de iki farklı şekilde yapılır.

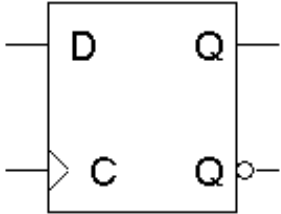
JK ters karakteristik tablosu (excitation table)

- Bir **ters karakteristik tablo (excitation table)** flip-flop'da istenilen durum değişikliğinin yapılabilmesi için girişlerinin ne olması gerektiğini gösterir. Bir başka deyişle, karakteristik tabloya tersten bakıştır:

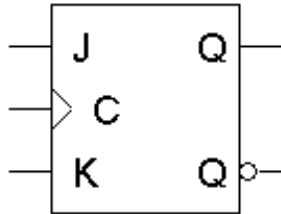
Q(t)	Q(t+1)	J	K	İşlem
0	0	0	x	Değişmez/Reset
0	1	1	x	Set/Tümlleme
1	0	x	1	Reset/Tümlleme
1	1	x	0	Değişmez/Set

J	K	Q(t+1)	İşlem
0	0	Q(t)	Değişmez
0	1	0	Reset
1	0	1	Set
1	1	Q'(t)	Tümlleme

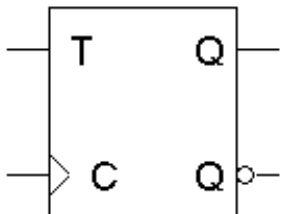
Tüm flip-flop'lar için ters karakteristik tabloları



Q(t)	Q(t+1)	D	İşlem
0	0	0	Reset
0	1	1	Set
1	0	0	Reset
1	1	1	Set



Q(t)	Q(t+1)	J	K	İşlem
0	0	0	x	Değişmez/Reset
0	1	1	x	Set/Tümlleme
1	0	x	1	Reset/Tümlleme
1	1	x	0	Değişmez /Set



Q(t)	Q(t+1)	T	İşlem
0	0	0	Değişmez
0	1	1	Tümlleme
1	0	1	Tümlleme
1	1	0	Değişmez

Örneğe geri dönersek...

- JK ters karakteristik tablosu kullanılarak *her bir* flip-flop'un girişleri bulunur. (şimdiki ve gelecek durumlara bakarak)

Q(t)	Q(t+1)	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

Şimdiki durum		Giriş X	Gelecek durum		Flip flop girişleri				Çıkış Z
Q ₁	Q ₀		Q ₁	Q ₀	J ₁	K ₁	J ₀	K ₀	
0	0	0	0	0	0	x	0	x	0
0	0	1	0	1	0	x	1	x	0
0	1	0	1	0	1	x	x	1	0
0	1	1	0	1	0	x	x	0	0
1	0	0	1	1	x	0	1	x	0
1	0	1	0	1	x	1	1	x	0
1	1	0	0	0	x	1	x	1	0
1	1	1	0	1	x	1	x	0	1

Adım 4: FF giriş ve çıkışları için denklemlerin bulunması

- Karnaugh diyagramı yardımıyla dört flip-flop'un her biri için girişlere ve çıkışa ait denklemler bulunur.
- Bu denklemler şimdiki durum ve girişler cinsindendir.
- JK FF kullanmanın avantajı: birkaç tane "don't care" durumuna sahip olmalarıdır. Bu sayede daha basit denklemler elde edilir.

Şimdiki durum		Giriş X	Gelecek durum		Flip flop girişleri				Çıkış Z
Q ₁	Q ₀		Q ₁	Q ₀	J ₁	K ₁	J ₀	K ₀	
0	0	0	0	0	0	x	0	x	0
0	0	1	0	1	0	x	1	x	0
0	1	0	1	0	1	x	x	1	0
0	1	1	0	1	0	x	x	0	0
1	0	0	1	1	x	0	1	x	0
1	0	1	0	1	x	1	1	x	0
1	1	0	0	0	x	1	x	1	0
1	1	1	0	1	x	1	x	0	1

$$J_1 = X' Q_0$$

$$K_1 = X + Q_0$$

$$J_0 = X + Q_1$$

$$K_0 = X'$$

$$Z = Q_1 Q_0 X$$

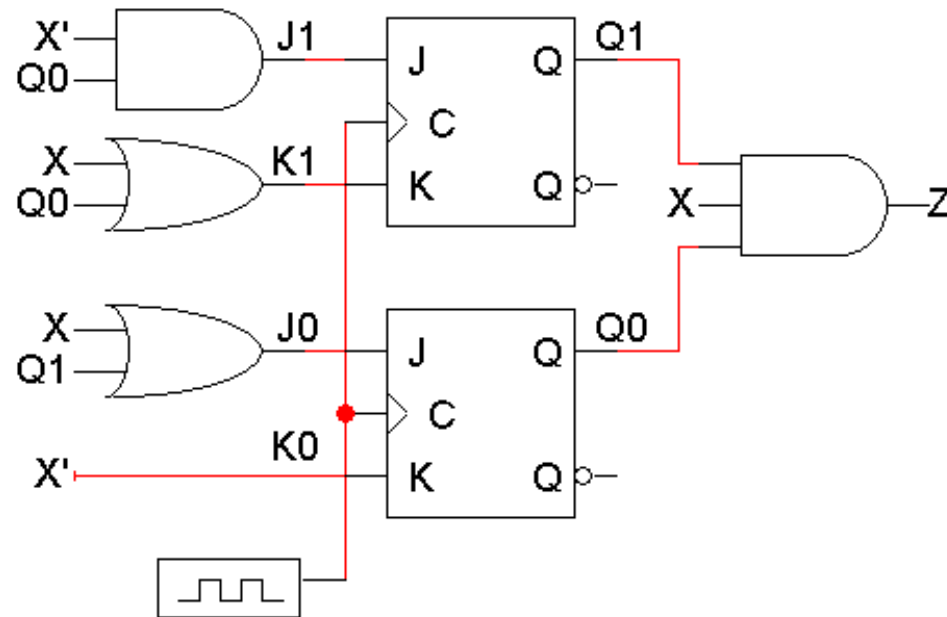
Adım 5: Devrenin kurulumu

- Son olarak, bu basitleştirilmiş denklemler temel alınarak devre kurulur.

$$J_1 = X' Q_0$$
$$K_1 = X + Q_0$$

$$J_0 = X + Q_1$$
$$K_0 = X'$$

$$Z = Q_1 Q_0 X$$



Aynı tasarımı D flip-flop ile gerçekleştirmek

- D FF'un tek girişi olduğu için tablo biraz daha basit olacak:

Şimdiki durum		Giriş X	Gelecek durum		Flip-flop girişleri		Çıkış Z
Q ₁	Q ₀		Q ₁	Q ₀	D ₁	D ₀	
0	0	0	0	0			0
0	0	1	0	1			0
0	1	0	1	0			0
0	1	1	0	1			0
1	0	0	1	1			0
1	0	1	0	1			0
1	1	0	0	0			0
1	1	1	0	1			1

D flip-flop giriş değerleri (Adım 3)

- D FF ters karakteristik tablosu:

Q(t)	Q(t+1)	D	İşlem
0	0	0	Reset
0	1	1	Set
1	0	0	Reset
1	1	1	Set

Şimdiki durum		Giriş X	Gelecek durum		Flip flop girişleri		Çıkış Z
Q ₁	Q ₀		Q ₁	Q ₀	D ₁	D ₀	
0	0	0	0	0	0	0	0
0	0	1	0	1	0	1	0
0	1	0	1	0	1	0	0
0	1	1	0	1	0	1	0
1	0	0	1	1	1	1	0
1	0	1	0	1	0	1	0
1	1	0	0	0	0	0	0
1	1	1	0	1	0	1	1

Denklemlerin bulunması (Adım 4)

- Karnaugh diyagramı yardımıyla aynı denklemleri bulabiliriz:

$$D_1 = Q_1 Q_0' X' + Q_1' Q_0 X'$$

$$D_0 = X + Q_1 Q_0'$$

$$Z = Q_1 Q_0 X$$

Devrenin kurulumu (Adım 5)

