

## 5. ARDIŞIL LOJİK DEVRELER

### 5.1. Ardışıl Lojik Devre Temelleri

SR Tutucu

Flip-Flop(FF)

Saat, Kenar tetikleme

D FF, JK FF, T FF

### 5.2. Ardışıl Devrelerin Analizi

Moore modeli: Çıkışlar=  $f(\text{şimdiki durum})$

Mealy modeli: Çıkışlar:  $f(\text{şimdiki durum, girişler})$

Durum Diyagramları

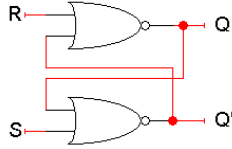
### 5.3. Ardışıl Devrelerin Tasarımı



- **Bellek nedir?** Bir bellek şu üç önemli özelliği sağlamalıdır:
  1. Bir değeri *tutabil*melidir (saklayabilmelidir).
  2. Biz belleğe kaydedilen değeri *okuyabil*miyiz.
  3. Biz belleğe kaydedilen değeri *değiştirebil*miyiz.

## SR Tutucu (SR Latch)

- VEYA DEĞİL kapıları ile tasarlanmış olan şu basit devreyi inceleyelim. **SR tutucu** devresinde iki tane giriş vardır: S ve R. ve bunlar Q ve Q' olmak üzere iki tane olan çıkışları kontrol etmektedirler:



- Burada Q ve Q' çıkışları devreye tekrar girmektedir. Yani sadece çıkış değil aynı zamanda giriştirler!
- Q ve Q' nün nasıl değiştiğini incelemek için sadece S ve R girişlerini ele almak yeterli değildir aynı zamanda Q ve Q' nün *şimdiki* değerlerine de bakmak gerekir:

$$Q_{\text{gelecek}} = (R + Q'_{\text{şimdiki}})'$$

$$Q'_{\text{gelecek}} = (S + Q_{\text{şimdiki}})'$$

- Değişik giriş değerlerinin bunları nasıl etkilediğine bakalım:

## SR = 00

- Eğer S = 0 ve R = 0 ise?

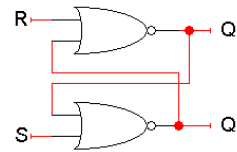
- Denklemler:

$$Q_{\text{gelecek}} = (0 + Q'_{\text{şimdiki}})' = Q_{\text{şimdiki}}$$

$$Q'_{\text{gelecek}} = (0 + Q_{\text{şimdiki}})' = Q'_{\text{şimdiki}}$$

- O halde SR = 00 ise,  $Q_{\text{gelecek}} = Q_{\text{şimdiki}}$  olmaktadır. i Q çıkışının değeri ne ise değişmez, aynı kalır.

- Bu durum tutucu içinde değer saklamaya karşılık gelir.

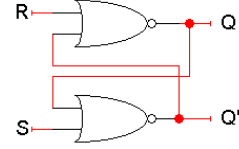


$$Q_{\text{gelecek}} = (R + Q'_{\text{şimdiki}})'$$

$$Q'_{\text{gelecek}} = (S + Q_{\text{şimdiki}})'$$

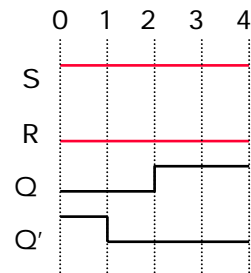
## SR = 10

- Eğer  $S = 1$  ve  $R = 0$  ise?
- $R = 0$  olduğundan,  $Q_{\text{gelecek}} = (0 + Q'_{\text{şimdiki}})' = Q_{\text{şimdiki}}$  (Ne ise ona devam)
- Ancak,  $S = 1$  olduğundan,  $Q'_{\text{gelecek}} = 0$ , ( $Q_{\text{şimdiki}}$ 'den bağımsız olarak)  $Q'_{\text{gelecek}} = (1 + Q_{\text{şimdiki}})' = 0$
- $Q'$  nün yeni değeri  $R=0$  ile birlikte üstteki VEYA DEĞİL kapısına gelir.  $Q_{\text{gelecek}} = (0 + 0)' = 1$
- O halde,  $SR = 10$  ise,  $Q'_{\text{gelecek}} = 0$  ve  $Q_{\text{gelecek}} = 1$
- Bu durum tutucuyu 1'e **set** etmeye karşılık gelir. Bir anlamda S girişi "set" yeridir.
- Not: Bu işlemde iki adım dolayısıyla da iki birim kapı gecikmesi söz konusudur. (S nin 1 olması ile  $Q_{\text{gelecek}}$  'in 1 olması arasında geçen süre)
- Fakat bir kez  $Q_{\text{gelecek}}$  1 olunca, çıkışların değişimi duracaktır. Bu bir **kararlı durum**dur. (**stable state**)



$$Q_{\text{gelecek}} = (R + Q'_{\text{şimdiki}})'$$

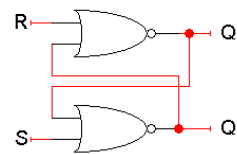
$$Q'_{\text{gelecek}} = (S + Q_{\text{şimdiki}})'$$



(Başlangıçta:  $Q=0$   $Q'=1$ )

## SR = 01

- Eğer  $S = 0$  ve  $R = 1$  ise?
- $R = 1$  olduğundan,  $Q_{\text{gelecek}} = 0$  olacaktır: ( $Q_{\text{şimdiki}}$ 'den bağımsız olarak)
- $Q_{\text{gelecek}} = (1 + Q'_{\text{şimdiki}})' = 0$
- $Q'$  nün yeni değeri alttaki VEYA DEĞİL kapısına gelir. Orada  $S = 0$  olduğundan:
- $Q'_{\text{gelecek}} = (0 + 0)' = 1$
- Böylece  $SR = 01$  ise,  $Q_{\text{gelecek}} = 0$  ve  $Q'_{\text{gelecek}} = 1$  olmaktadır.
- Bu durum tutucuyu 0'a **resetleme** (veya **clear**) işlemine karşılık gelir. Bir anlamda R girişi "reset" yeridir.
- Yine, R ile birlikte  $Q'_{\text{gelecek}}$  'deki değişimin gerçekleşmesi için iki kapı gecikmesi söz konusudur.



$$Q_{\text{gelecek}} = (R + Q'_{\text{şimdiki}})'$$

$$Q'_{\text{gelecek}} = (S + Q_{\text{şimdiki}})'$$

## SR tutucu'ları ...

S	R	Q
0	0	Değişmez
0	1	0 (reset)
1	0	1 (set)

- SR tutucu'ları birer bellektir!
- Yandaki tablo tutucuların bir belleğin sağlanması gerekli olan tüm özellikleri sağladığını göstermektedir. Set edilebilir, resetlenebilir ve ayrıca üzerindeki değeri tutabilir.
- Q çıkışı tutucuda tutulmuş olan veriyi gösterir. Aynı zamanda tutucu için **durum** olarak da adlandırılır.
- Bu tablo **durum tablosu** şeklinde genişletilebilir. Durum tablosunda Q ve Q' nün *şimdiki* değerleri ve S ve R giriş değerleri ile birlikte bunlara bağlı olarak hesaplanmış olan Q ve Q' nün *gelecek* değerleri yer almaktadır.

Giriş		Şimdiki		Gelecek	
S	R	Q	Q'	Q	Q'
0	0	0	1	0	1
0	0	1	0	1	0
0	1	0	1	0	1
0	1	1	0	0	1
1	0	0	1	1	0
1	0	1	0	1	0

- SR tutucu'ları ardışıldır!
- Dikkat edilirse  $SR = 00$  iken, Q nun gelecek değeri Q nun şimdiki değerine bağlı olarak 0 da olabilir 1 de olabilir
- O halde daha önce de belirtildiği gibi aynı giriş değerleri farklı çıkış değerlerine sebep olabilmektedir.
- Ardışıl devrelerin bu özelliği kombinezonsal devrelere göre fark yaratmaktadır.

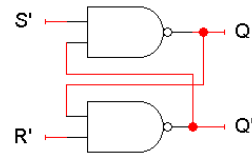
## SR = 11 olursa?

- Hem  $Q_{\text{gelecek}}$  ve hem de  $Q'_{\text{gelecek}}$  0 olacak!
- Ancak bu durum Q ve Q' nün her zaman birbirinin tümleyeni olması gerçeğine ters düşmektedir.
- Ayrıca, bir adım daha ilerletirsek de, bu durum düzelmiyor:  
 $Q_{\text{gelecek}} = (1+0)' = 0$   
 $Q'_{\text{gelecek}} = (1+0)' = 0$
- Sonuç olarak hiçbir zaman  $SR=11$  olmamalı!

S	R	Q
0	0	Değişmez
0	1	0 (reset)
1	0	1 (set)
1	1	Kullanılmaz!

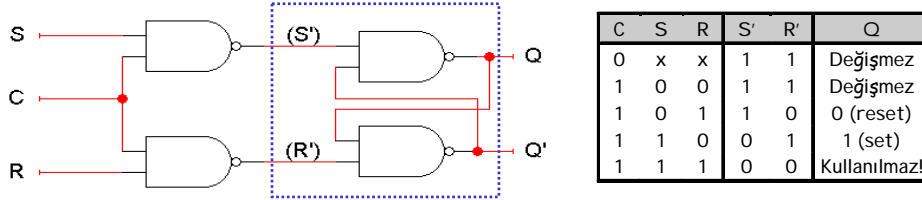
## S'R' tutucu

- Tutucuların çeşitli tipleri vardır.
- VEYA DEĞİL yerine VE DEĞİL kapıları kullanılırsa **S'R'** tutucu elde edilir.



S'	R'	Q
1	1	Değişmez
1	0	0 (reset)
0	1	1 (set)
0	0	Kullanılmaz!

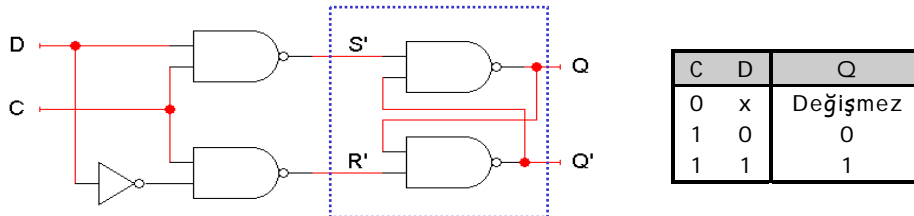
## SR tutucu için kontrol girişi



- Kontrol girişinin işlevi izin (enable) girişine benzemektedir.
- Kesikli çizgiyle belirtilen kutunun içi S'R' tutucudur!  
Eklenen iki tane VE DEĞİL kapısı ise, bu S'R' tutucunun girişlerini denetler.

## D tutucu

- Son olarak göreceğimiz D tutucunun temelinde S'R' tutucu vardır. Eklenmiş olan kapılar S' ve R' sinyallerini üretir. D: "data" ve C "control".
  - C = 0 ise, S' ve R' nin ikisi de 1 dir ve Q durumu değişmez.
  - C = 1 ise, tutucunun Q çıkışı D girişine eşit olur.
- Set ve reset girişi gibi karışıklıklar yoktur.



- Ayrıca, bu tutucu tipinde kaçınılması gereken "kötü" girişler söz konusu değildir. C ve D'ye ait dört kombinasyon da geçerlidir.

## Flip-Flop

- Bir devrede bellek elemanı olarak kullanılmak üzere tutucuları inceledik.
- Tutucular bazı problemlere sahiptir:
  - Tutucuya ne zaman izin vereceğimizi (enable) bilmeliyiz.
  - Tutucuyu çabuk devre dışı (disable) bırakabilmeliyiz.
  - Bir başka deyişle, büyük devrelerde tutucuların zamanlamasını kontrol etmek zordur.
- Bu problemleri saat işareti ve Flip-Flop elemanı ile çözmek mümkündür!
  - Saat işareti belleğe yazacağımız zamanı bildirir.
  - Flip-Flop ise, kesin olarak belirlenen zamanda belleğe yazma işlemini gerçekleştirmeyi sağlar.
  - Bu ikisi birlikte kullanılırsa, bellek zamanlaması problemini göz ardı ederek devrelerimizi tasarlayabiliriz.

## Saat (Clock) ve Senkronizasyon

- **Saat** çıkışı sürekli olarak belirli bir periyot ile 0 ve 1 arasında değişen özel bir devre elemanıdır.

saat periyodu

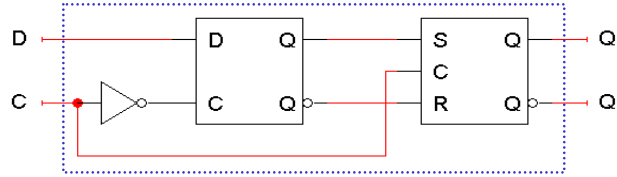


- Saat'in 1'den 0'a değişmesi ile başlayan ve tekrar 1 oluncaya kadar geçen süreye **saat periyodu**, veya **saat devir süresi** denilir.
- **Saat frekansı** saat periyotunun tersidir. Birimi ise **hertz** dir.
- Saatler genellikle devrelerin senkronizasyonu için kullanılır. Devrelerde belli işlemlerin başlaması için tetikleme amaçlı kullanılırlar. Örneğin, tutucuya yazma işlemi gibi.
- Birden fazla devre aynı saati kullanırsa senkronizasyon sağlanmış olur. Bu, insanların senkronizasyon için saat kullanmalarına benzer bir durumdur.

## Flip-Flop

- D Flip-Flop (D FF)

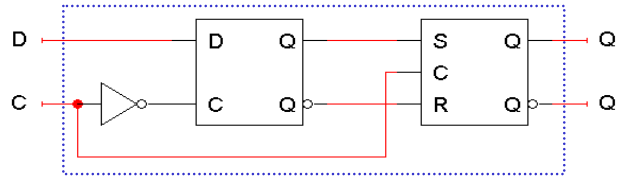
- Flip-Flop girişleri: C ve D, ve çıkışları: Q ve Q'
- Solda bir D tutucu: **ana (master)**, sağda bir SR latch: **uydu (slave)**



- Not:

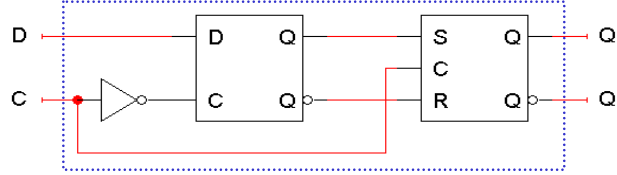
- Flip-Flop D girişi doğrudan ana tutucuya bağlıdır.
- Ana tutucunun çıkışları uydunun girişleridir.
- Flip-Flop çıkışı doğrudan uydu tutucunun çıkışlarıdır.

## D Flip-Flop'da C=0 ise,



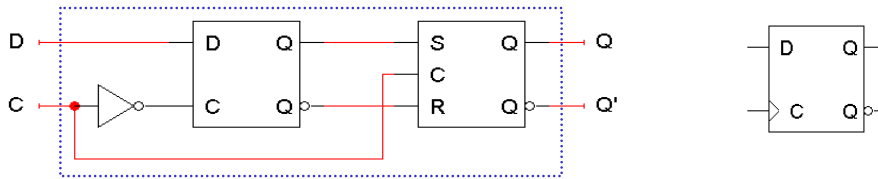
- D Flip-Flop C kontrol girişi ya D tutucuya izin verir yada SR tutucuya izin verir. İkisinin aynı anda çalışmasına izin vermesi mümkün değildir.
- C = 0 ise:
  - Ana tutucunun çalışmasına izin verilir ve çıkışı Flip-Flop'un D girişi olur. Eğer D girişi değişirse ana tutucunun çıkışı da değişir.
  - Uydu tutucunun çalışmasına izin verilmediği için D tutucunun çıkışından etkilenmez. Bu durumda uydu tutucunun çıkışı, Flip-Flop'un çıkışı Flip-Flop'un şimdiki durumunu gösterir.

### D Flip-Flop'da $C=1$ ise,



- Ardından  $C = 1$  olur olmaz, (saatin çıkan kenarında,  $C=0 \rightarrow 1$ )
  - Ana tutucunun çalışmasına izin verilmez. Çıkışı da  $C=1$  olmadan önceki son D giriş değeri olarak kalır.
  - Ana tutucunun çalışmasına izin verilmediği için D girişindeki değişiklikler  $C = 1$  olduğu sürece ana tutucuyu etkilemez.
  - Uydu tutucunun çalışmasına izin verilir. S ve R değerlerine göre uydu tutucunun çıkışı ve dolayısıyla Flip-Flop'un çıkışı belirlenir.

### Çıkan Kenar Tetikleme

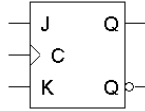


- Bu bir **çıkan kenar tetiklemeli** Flip-Flop'dur. Flip-Flop'un Q çıkışı sadece C girişinin çıkan kenarında değişebilir.
- D Flip-Flop'unun davranışı, Q nun çıkan kenarda değişmesi dışında D tutucuyla aynıdır.

C	D	Q
0	x	Değişmez
1	0	0 (reset)
1	1	1 (set)

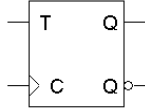
## Flip-flop çeşitleri

- D Flip-Flop temel alınarak çeşitli Flip-Flop lar tasarlanmıştır.
- JK Flip-Flop** (SR'ye benziyor ama JK=11 Flip-Flop un şimdiki durumunun tümleyenini almak için kullanılır.)



C	J	K	Q <sub>gelecek</sub>
0	x	x	Değişmez
1	0	0	Değişmez
1	0	1	0 (reset)
1	1	0	1 (set)
1	1	1	Q' <sub>şimdiki</sub>

- T Flip-Flop** sadece şimdiki durumu tutar veya tümleyenini alır.



C	T	Q <sub>gelecek</sub>
0	x	Değişmez
1	0	Değişmez
1	1	Q' <sub>şimdiki</sub>

## Karakteristik tablolar ve denklemler

- Her Flip-Flop için bir **karakteristik tablo** vardır. Bu tablolar giriş ve şimdiki duruma  $Q(t)$  bağlı olarak gelecek durumu  $Q(t+1)$  gösterirler. (Basitleştirmek için kontrol girişi C tablolara alınmamıştır. Ayrıca, aksi söylenmedikçe Flip-Flop lar çıkan kenar tetiklemelidir.)
- karakteristik denklemler**,  $Q(t+1)$  gelecek durum,  $Q(t)$  şimdiki durum ve giriş cinsinden bir fonksiyon ile ifade edilir.

D	Q(t+1)	İşlem
0	0	Reset
1	1	Set

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

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

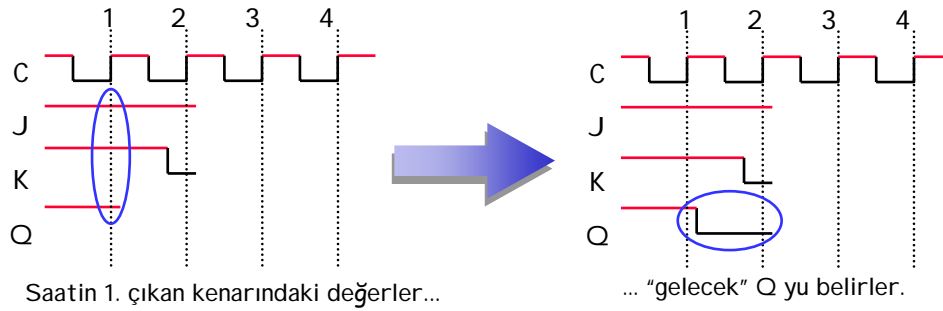
$$Q(t+1) = K'Q(t) + JQ'(t)$$

T	Q(t+1)	İşlem
0	Q(t)	Değişmez
1	Q'(t)	Tümleme

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

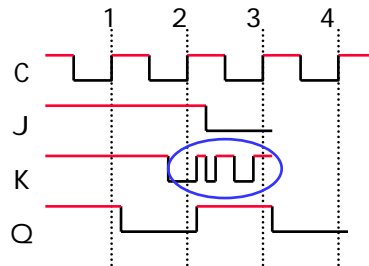
## Flip flop zamanlama diyagramları

- JK Flip-Flop için zamanlama diyagramı örneği: C saatinin 1. çıkan kenarında  $J=1$ ,  $K=1$  ve  $Q(1) = 1$  dir. O halde gelecek durum  $Q(2) = Q(1)' = 0$  oluyor.
- Saatin 1. çıkan kenarının hemen ardından  $Q(2)=0$  değerini alır. Artık bu değer şimdiki durumdur ve saatin 2. çıkan kenarına kadar değişmez.



## Önemli Not

- Eğer Flip-Flop çıkan kenar tetiklemeli ise, çıkışlar sadece girişlerin çıkan kenarındaki değerlerine göre değişir. Aşağıdaki diyagramda bu konu incelenmektedir:
  - K ikinci ve üçüncü çıkan kenarlar arasında birden fazla kere değişmektedir.
  - Ancak, K'daki bu değişimden bağımsız olarak, üçüncü çıkan kenardaki giriş değerleri olan  $J=0$ ,  $K=1$  ve  $Q=1$  değerleri bir sonraki durumu etkileyecek ve Q nun değeri 0 olarak değişecektir.

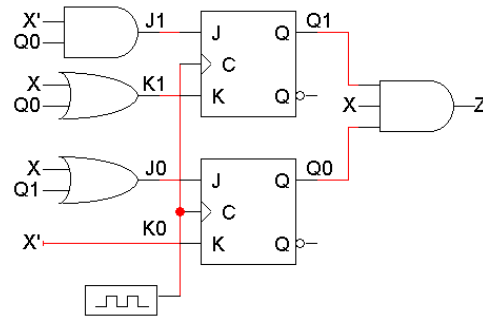


## 5.2. Ardışıl Devre Analizi (Sequential Circuit Analysis)

- Mealy Modeli / Moore Modeli ???
- Bir ardışıl devrenin çıkışı iki farklı şekilde ifade edilebilir:
  - Moore modeli: Çıkışlar=  $f(\text{şimdiki durum})$
  - Mealy modeli: Çıkışlar:  $f(\text{şimdiki durum, girişler})$



## Bir ardışıl devre örneği

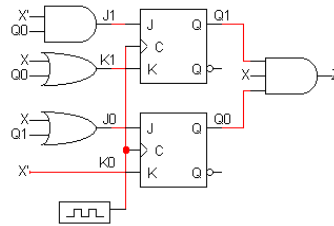


- İki tane JK Flip-Flop içeren bir ardışıl devre
- Durum veya Bellek:  $Q1Q0$
- Bir giriş: X; Bir çıkış: Z

## Bir ardışıl devre nasıl tanımlanır?

- Kombinezonsal devre - Doğruluk tablosu (Girişler ile çıkışlar arasındaki ilişkiyi gösterir)
- Ardışıl devre - **Durum tablosu** (Girişler ile şimdiki durumların *solda*, çıkışlar ile gelecek durumların ise *sağda* olduğu tablodur.)
  - Şimdiki durumlar ve girişler kullanılarak FF ların gelecek durumlarının bulunması gerekir
  - Devrenin çıkışına ait fonksiyonun bulunması gerekir:
    - > Moore modeli devresi için şimdiki durumlar cinsinden
    - > Mealy modeli devresi için şimdiki durumlar ve girişler cinsinden

## Örnek devrenin durum tablosu

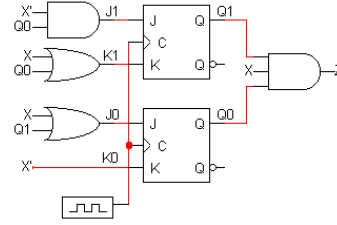


Şimdiki Durum		Giriş	Gelecek Durum		Çıkış
Q <sub>1</sub>	Q <sub>0</sub>	X	Q <sub>1</sub>	Q <sub>0</sub>	Z
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

- Çıkışları yazmak kolay! Devre diyagramından bakılırsa:

$$Z = Q_1 Q_0 X$$

Mealy modeli devresi !!!



Şimdiki Durum		Giriş	Gelecek Durum		Çıkış
Q <sub>1</sub>	Q <sub>0</sub>	X	Q <sub>1</sub>	Q <sub>0</sub>	Z
0	0	0			0
0	0	1			0
0	1	0			0
0	1	1			0
1	0	0			0
1	0	1			0
1	1	0			0
1	1	1			1

## Flip-flop girişleri

- Gelecek durumları bulmak kolay değildir:

### Adım 1:

Flip-flop girişleri için Boole fonksiyon ifadeleri bulunur.

### Adım 2:

Bu fonksiyon ifadeleri kullanılarak gerçek Flip-Flop giriş değerleri bulunur. (şimdiki durum ve girişler üzerinden olası bütün kombinasyonlar için)

Bunun için durum tablosunda arada yeni kolonlar oluşturulur.

### Adım 3:

Flip-Flop karakteristik tablosu veya denklemler kullanılarak gelecek durum değerleri bulunur. (şimdiki durumlar ve girişlere bakılarak elde edilmiş Flip-Flop giriş değerleri kullanılır)

## Adım 1: Flip-flop giriş denklemleri

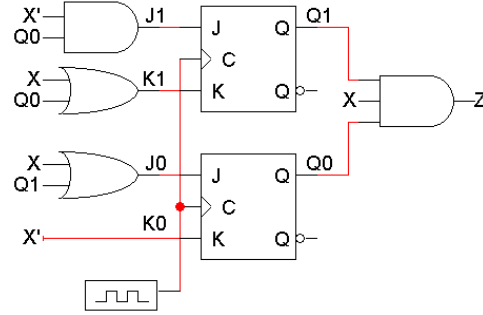
- Bu örnek için, **Flip-Flop giriş denklemleri:**

$$J_1 = X' Q_0$$

$$K_1 = X + Q_0$$

$$J_0 = X + Q_1$$

$$K_0 = X'$$



## Adım 2: Flip-flop giriş değerleri

- Bu denklemler kullanılarak, şimdiki durumlar  $Q_1Q_0$  ve giriş  $X$ 'in tüm kombinasyonları için  $J_1$ ,  $K_1$ ,  $J_0$  ve  $K_0$  hesaplanarak tabloya yazılır.

$$J_1 = X' Q_0$$

$$K_1 = X + Q_0$$

$$J_0 = X + Q_1$$

$$K_0 = X'$$

Şimdiki Durum		Giriş	Flip-flop Girişleri			
$Q_1$	$Q_0$	$X$	$J_1$	$K_1$	$J_0$	$K_0$
0	0	0	0	0	0	1
0	0	1	0	1	1	0
0	1	0	1	1	0	1
0	1	1	0	1	1	0
1	0	0	0	0	1	1
1	0	1	0	1	1	0
1	1	0	1	1	1	1
1	1	1	0	1	1	0

### Adım 3: Gelecek durumlar

- Son olarak, JK Flip-Flop karakteristik tablosu veya denklemleri yardımıyla, şimdiki durum ve giriş değerlerine bakılarak her Flip-Flop çıkışı için gelecek durum bulunur.

- JK Flip-Flop'un karakteristik denklemi:

$$Q(t+1) = K'Q(t) + JQ'(t)$$

- Bu örnekteki iki tane JK Flip-Flop lar için karakteristik denklemler:

$$Q_1(t+1) = K_1'Q_1(t) + J_1Q_1'(t)$$

$$Q_0(t+1) = K_0'Q_0(t) + J_0Q_0'(t)$$

- Gelecek durumu hesaplamak için karakteristik tablo da kullanılabilir

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

$$Q_1(t+1) = K_1'Q_1(t) + J_1Q_1'(t)$$

$$Q_0(t+1) = K_0'Q_0(t) + J_0Q_0'(t)$$

Şimdiki Durum		Giriş X	FF Girişler				Gelecek Durum	
Q <sub>1</sub>	Q <sub>0</sub>		J <sub>1</sub>	K <sub>1</sub>	J <sub>0</sub>	K <sub>0</sub>	Q <sub>1</sub>	Q <sub>0</sub>
0	0	0	0	0	0	1	0	0
0	0	1	0	1	1	0	0	1
0	1	0	1	1	0	1	1	0
0	1	1	0	1	1	0	0	1
1	0	0	0	0	1	1	1	1
1	0	1	0	1	1	0	0	1
1	1	0	1	1	1	1	0	0
1	1	1	0	1	1	0	0	1

## Durum tablosunun oluşumu

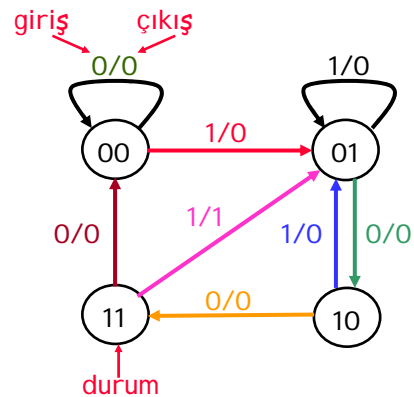
- Başlangıçta tabloda: **Şimdiki Durum** ve **Girişler**
  - Şimdiki Durum** ve **Girişler** → FF Girişleri
  - Şimdiki Durum** ve FF Girişleri → **Gelecek Durum** (Flip-Flop çalışma tablosundan)
  - Şimdiki Durum** ve **Girişler** → **Çıkışlar**

Şimdiki Durum		Giriş X	FF Girişleri				Gelecek Durum		Çıkış Z
Q <sub>1</sub>	Q <sub>0</sub>		J <sub>1</sub>	K <sub>1</sub>	J <sub>0</sub>	K <sub>0</sub>	Q <sub>1</sub>	Q <sub>0</sub>	
0	0	0	0	0	0	1	0	0	0
0	0	1	0	1	1	0	0	1	0
0	1	0	1	1	0	1	1	0	0
0	1	1	0	1	1	0	0	1	0
1	0	0	0	0	1	1	1	1	0
1	0	1	0	1	1	0	0	1	0
1	1	0	1	1	1	1	0	0	0
1	1	1	0	1	1	0	0	1	1

## Durum Diyagramları (Mealy modeli)

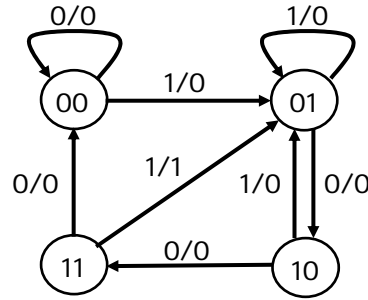
- Durum diyagramı çizerek durum tablosunu grafiksel olarak ifade edebiliriz.
- Bu örneğe ait durum diyagramı:

Şimdiki Durum		Giriş X	Gelecek Durum		Çıkış Z
Q <sub>1</sub>	Q <sub>0</sub>		Q <sub>1</sub>	Q <sub>0</sub>	
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

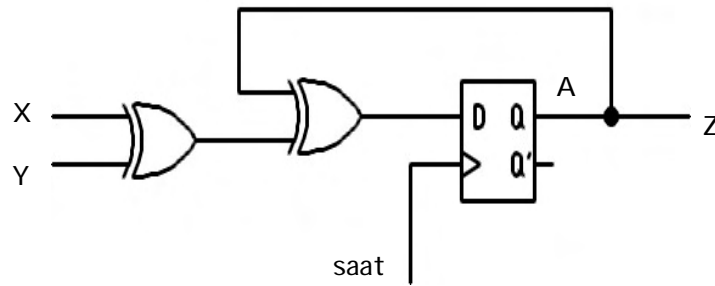


### Durum diyagramlarının boyutu

- Durum diyagramı kontrolü:
  - Eğer  $n$  tane Flip-Flop varsa, diyagramda  $2^n$  tane durum noktası olmalı
  - Eğer  $m$  tane giriş varsa, her bir durum noktasında  $2^m$  tane çıkış oku olmalı
- Bu örnekte,
  - İki tane Flip-Flop var; Dört tane durum noktası olmalı.
  - Bir tane giriş var; Her bir durum noktasında iki tane çıkış oku olmalı.

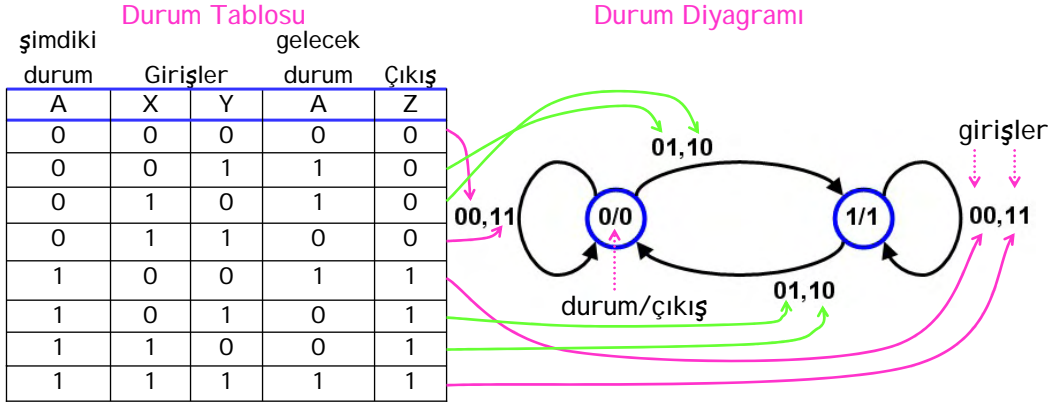
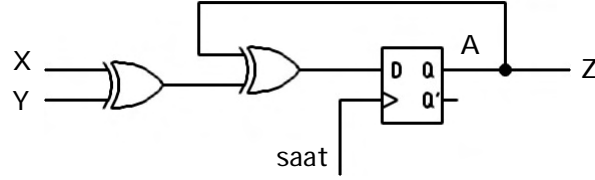


### Moore modeli devresi



- İki giriş: X ve Y; Bir çıkış: Z
- Bir durum: A
- Dikkat edilirse  $Z = A$ , yani sadece şimdiki duruma bağlı bir fonksiyondur.

## Durum Tablosu ve Durum Diyagramı (Moore modeli)



## Ardışıl devre analizi - Özet

- Bir ardışıl devrenin analizini yapmak için:
  - Devrenin çıkışına ve Flip-Flop'ların girişine ait Boole fonksiyonu ifadeleri bulunur.
  - Bu ifadeler kullanılarak durum tablosundaki çıkış ve Flip-Flop girişi kolonlarına ilgili değerler yazılır.
  - Son olarak, ilgili Flip-Flop'a ait karakteristik tablo veya denklem kullanılarak tablodaki gelecek durum kolonları yazılır.
- Ardışıl devre analizinin sonucu ya bir durum tablosu yada bir durum diyagramı olacaktır.

### 5.3. Ardışıl Devre Sentezi (Sequential Circuit Design)

Ardışıl devre tasarım adımları:

**Adım 1:**

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

**Adım 2:**

Durum tablosundaki durumlara ikili kod atanır (Eğer henüz atanmamışsa). N tane durum varsa, ikili kodlar en az  $\lceil \log_2 n \rceil$  digit olmalıdır ve devrede en az  $\lceil \log_2 n \rceil$  Flip-Flop olacaktır.

**Adım 3:**

Her bir Flip-Flop ve durum tablosundaki her bir satır için, gelecek durumları belirleyebilmek amacıyla, Flip-Flop giriş değerleri bulunur. Bunun için Flip-Flop çalışma 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.
- Yakalayıcı devrenin tek bir girişi vardır: X.
  - Girişin her bir biti bir saat periyodunda sağlanır.
  - Bu uzun giriş dizilerinin uygulanması için en kolay yoldur.
- Tek bir çıkış vardır: Z. İstenen bit dizisi bulunduğunda 1 çıkışını verir.
- **Örnek:** Aranacak olan bit dizisi: 1001

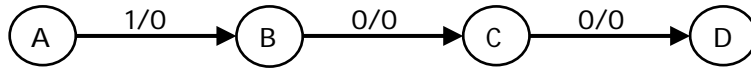
Girişler: 1 1 1 001 1 0 1 001 001 1 0 ...

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

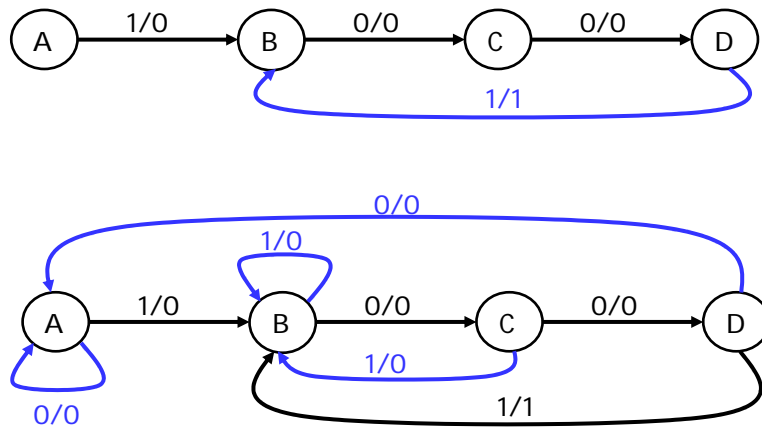
- Bu tasarım da bir ardışıl devre tasarlanması gerekmektedir. Çünkü devrenin diziyi yakalayabilmesi için birkaç önceki çıkışı da hatırlayabilmesi gerekmektedir.

## Adım 1: Durum tablosu yapma

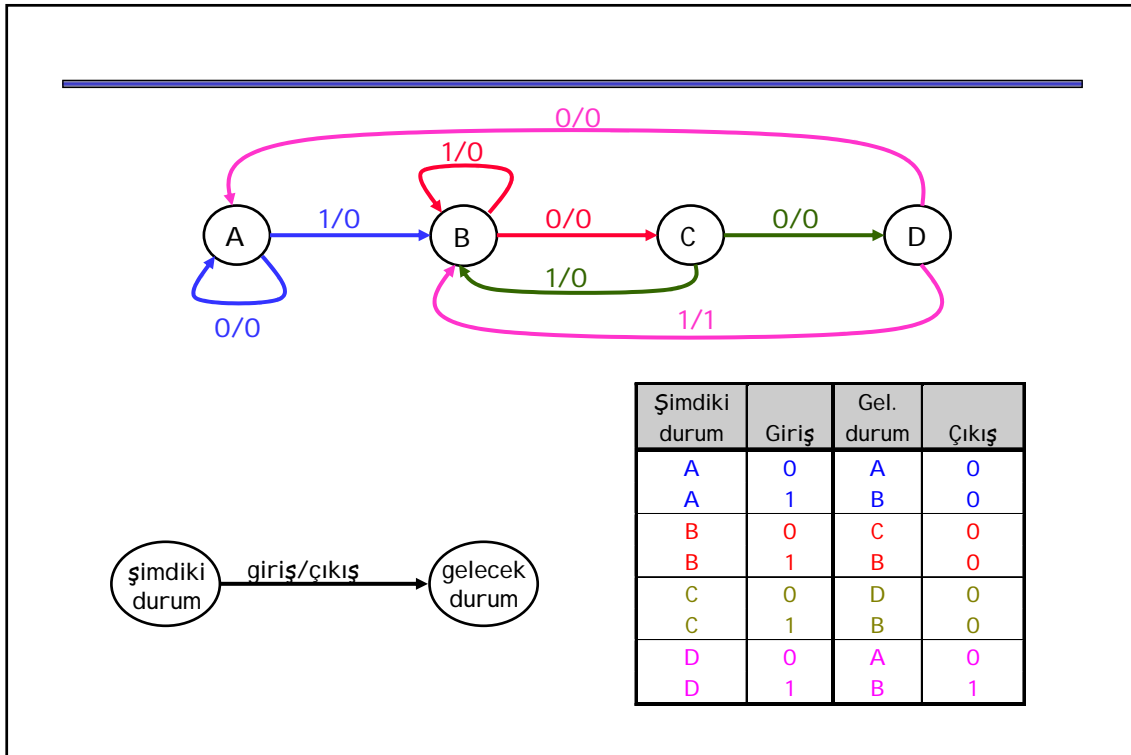
- Dizi yakalayıcısı için öncelikle durum diyagramı yapıp, ardından tabloyu oluşturalım:
  - Önceki saat periyotlarındaki girişleri hatırlamalı
  - Örneğin, eğer önceki üç giriş 100 ve şu anki giriş 1 ise, çıkış 1 olacaktır.
  - Genellikle, bu dizinin belli kısımlarının oluşumunu incelemek gerekiyor—bu örnekte: 1, 10, ve 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 durum noktası için *iki* çıkış oku vardır: X=0 ve X=1 girişi için.



## Adım 2: Durumlara ikili 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	Gelecek Durum		Flip flop girişleri				Çıkış
$Q_1$	$Q_0$	X	$Q_1$	$Q_0$	$J_1$	$K_1$	$J_0$	$K_0$	Z
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 ustalık ister. 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	$Q'(t)$	Tümleme

- 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 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.

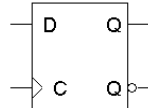
## JK karakteristik tablosu (excitation table)

- Bir **karakteristik tablo** Flip-Flop'ta istenilen durum değişikliğinin yapılabilmesi için girişlerinin ne olması gerektiğini gösterir. Bir başka deyişle, karakteristik tabloya tersten bakıştır:

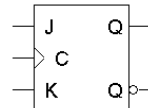
Q(t)	Q(t+1)	J	K	İşlem
0	0	0	x	Değişmez/Reset
0	1	1	x	Set/Tümleme
1	0	x	1	Reset/ Tümleme
1	1	x	0	Değişmez/Set

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

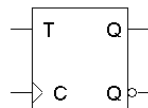
## Tüm Flip-Flop(FF)lar için karakteristik tablolar



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/tümleyen
1	0	x	1	Reset/tümleyen
1	1	x	0	Değişmez/set



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

### Örneğe geri dönersek...

- JK 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	Gelecek durum		Flip flop girişleri				Çıkış
Q <sub>1</sub>	Q <sub>0</sub>	X	Q <sub>1</sub>	Q <sub>0</sub>	J <sub>1</sub>	K <sub>1</sub>	J <sub>0</sub>	K <sub>0</sub>	Z
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 cinsindedir.
- JK FF kullanmanın avantajı: birkaç tane keyfi (x) duruma sahip olmalarıdır. Bu sayede daha basit denklemler elde edilir.

Şimdiki durum		Giriş	Gelecek durum		Flip flop girişleri				Çıkış
Q <sub>1</sub>	Q <sub>0</sub>	X	Q <sub>1</sub>	Q <sub>0</sub>	J <sub>1</sub>	K <sub>1</sub>	J <sub>0</sub>	K <sub>0</sub>	Z
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

$$J_1 = X' Q_0$$

$$K_1 = X + Q_0$$

$$J_0 = X + Q_1$$

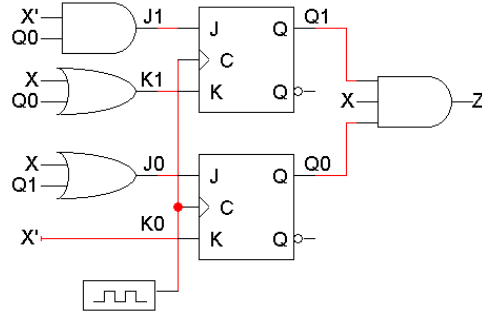
$$K_0 = X'$$

$$Z = Q_1 Q_0 X$$

### Adım 5: Devrenin kurulumu

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

$$\begin{aligned} J_1 &= X' Q_0 \\ K_1 &= X + Q_0 \\ J_0 &= X + Q_1 \\ K_0 &= X' \\ Z &= Q_1 Q_0 X \end{aligned}$$



### Aynı tasarımı D Flip-Flop ile gerçekleştirmek

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

Şimdiki durum		Giriş	Gelecek durum		Flip-flop girişleri		Çıkış
Q <sub>1</sub>	Q <sub>0</sub>	X	Q <sub>1</sub>	Q <sub>0</sub>	D <sub>1</sub>	D <sub>0</sub>	Z
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 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	Gelecek durum		Flip flop girişleri		Çıkış
Q <sub>1</sub>	Q <sub>0</sub>	X	Q <sub>1</sub>	Q <sub>0</sub>	D <sub>1</sub>	D <sub>0</sub>	Z
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)

