

6.ARDİŞİL LOJİK DEVRELER

6.1.Ardışıl Lojik Devre Temelleri

SR Tutucu

Flip-Flop(FF)

Saat, Kenar tetikleme

D FF, JK FF, T FF

6.2.Ardışıl Devrelerin Analizi

Moore modeli: Çıkışlar= f(şimdiki durum)

Mealy modeli: Çıkışlar: f(şimdiki durum, girişler)

Durum Diyagramları

6.3.Ardışıl Devrelerin Tasarımı

Y.Doç.Dr.Tuncay UZUN

6. Ardışıl Lojik Devreler

1



- **Bellek nedir?** Bir bellek şu üç önemli özelliği sağlamalıdır:
 1. Bir değeri *tutabilmelidir* (saklayabilmelidir).
 2. Biz belleğe kaydedilen değeri *okuyabilmeliyiz*.
 3. Biz belleğe kaydedilen değeri *değiştirebilmeliyiz*.

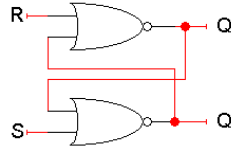
Y.Doç.Dr.Tuncay UZUN

6. Ardışıl Lojik Devreler

2

SR Tutucu (SR Latch)

- NOR 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:

Y.Doç.Dr.Tuncay UZUN

6. Ardışıl Lojik Devreler

3

SR = 00

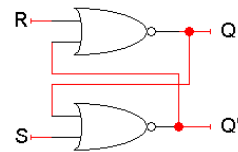
- Eğer S = 0 ve R = 0 ise?

- 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. Yani Q'nun değeri ne ise o aynen kalır.
- Bu durum tutucu içinde değer saklamaya karşılık gelir.



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

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

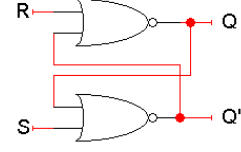
Y.Doç.Dr.Tuncay UZUN

6. Ardışıl Lojik Devreler

4

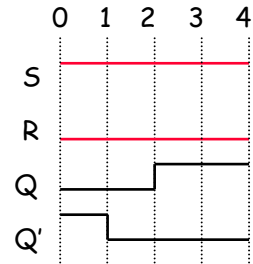
SR = 10

- 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' 'nin yeni değeri $R=0$ ile birlikte üstteki NOR 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_{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**)



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

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



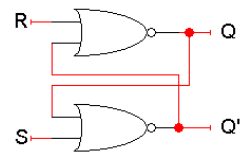
(Başlangıçta: $Q=0$ $Q'=1$)

Y.Doç.Dr.Tuncay UZUN

6. Ardışıl Lojik Devreler

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' 'nin yeni değeri alttaki NOR 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'_{gelecek} 'deki değişimin gerçekleşmesi için iki kapı gecikmesi söz konusudur.



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

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

Y.Doç.Dr.Tuncay UZUN

6. Ardışıl Lojik Devreler

6

SR tutucu'ları ...

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

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 birer bellektir!
- Yandaki tablo tutucuların bir belleğin sağlması 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.

- 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.

Y.Doç.Dr.Tuncay UZUN

6. Ardışıl Lojik Devreler

7

SR = 11 olursa?

S'R' tutucu

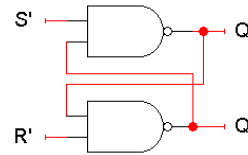
- Hem $Q_{gelecek}$ ve hem de $Q'_{gelecek}$ 0 olacak!
- Ancak bu durum Q ve Q' nün her zaman birbirinin tümleyeni olması gerçeğine ters düşmektedir.
- Ayrıca, bir adım daha ilerletirsek de, bu durum düzelmiyor:

$$= (1+0)' = 0 \quad Q'_{gelecek} =$$

$$(1+0)' = 0$$
- Sonuç olarak hiçbir zaman $SR=11$ olmamalı!

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

- Tutucuların çeşitli tipleri vardır.
- NOR yerine NAND kapıları kullanılırsa **S'R' tutucu** elde edilir.



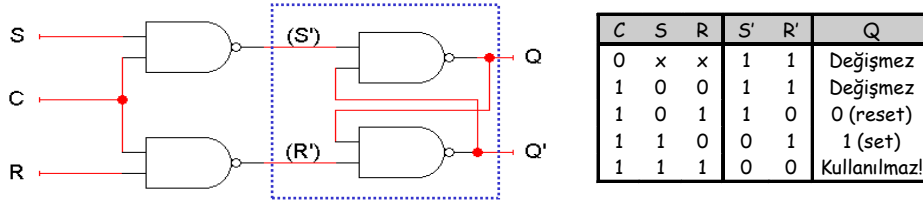
S'	R'	Q
1	1	Değişmez
1	0	0 (reset)
0	1	1 (set)
0	0	Kullanılmaz!

Y.Doç.Dr.Tuncay UZUN

6. Ardışıl Lojik Devreler

8

SR tutucu için kontrol girişi



- Kontrol girişinin işlevi enable girişine benzemektedir.
- Kesikli kutunun içi S'R' tutucu !
Eklenen iki tane NAND kapısı ise, bu S'R' latch'ine gelen girişleri ayarlıyor.

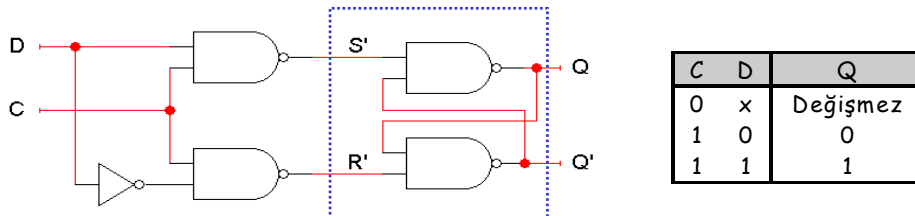
Y.Doç.Dr.Tuncay UZUN

6. Ardışıl Lojik Devreler

9

D tutucu

- Son olarak göreceğimiz D tutucu'nun 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, tutucu'nun 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.

Y.Doç.Dr.Tuncay UZUN

6. Ardışıl Lojik Devreler

10

Flip-Flop

- Bir devrede bellek elemanı olarak kullanılmak üzere tutucuları inceledik.
- Tutucular bazı problemlere sahiptir:
 - Tutucuyu ne zaman enable yapacağımızı bilmeliyiz.
 - Tutucuyu çabucak devredışı bırakabilmeliyiz (disable edebilmeliyiz).
 - Bir başka deyişle, büyük devrelerde latchlerin zamanlamasını kontrol etmek zordur.
- Bu problemleri iki yeni eleman ile çözmemiz mümkündür: saat ve flip-flop'lar
 - Saat, 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



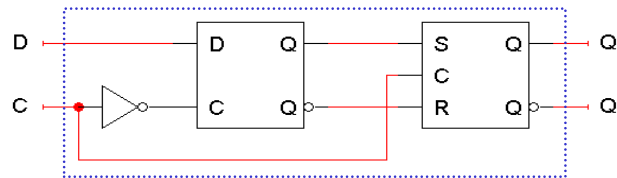
- Saat'in 1'den 0'a değişmesi ile başlayan ve tekrar 1 oluncaya kadar geçen süreye **saat periyodu**, veya **saat devir süresi** denilir.
- **Saat frekansı** saat periyotunun tersidir. Birimi ise **hertz** dir.
- Saatler genellikle devrelerin senkronizasyonu için kullanılır. Devrelerde belli işlemlerin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

Saat

- Saatler büyük olarak bilgisayar mimarisinde kullanılmaktadır.
- Tüm işlemciler bir iç saat ile çalışmaktadır.
 - Modern işlemciler (chip ler) 3.2 GHz'e kadar uzanan frekanslarda çalışmaktadır.
 - Bu da cycle time ı 0.31 ns kadar küçültmektedir!
- Dikkat... Daha yüksek frekans her zaman için daha hızlı makineye karşılık gelmez!
 - Her bir saat periyotunda ne kadar iş yapılabileceğine bakmak gerekir.
 - Ne kadar eleman 0.31 ns gibi sürede iş yapabilir?

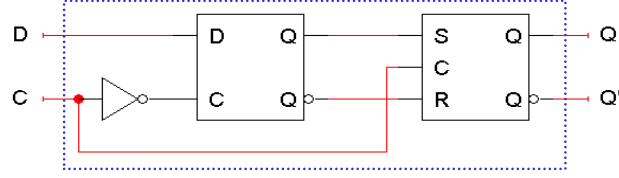
Flip-flop

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



- **Not:**
 - Flip-flop'un D girişi doğrudan master latche bağlı
 - Master latch in çıkışları slave e giriş oluyor. (master x slave: ana x uydu)
 - Flip-flop'un çıkışı doğrudan slave latch den alınıyor.

D flip-flop'da C=0 ise,



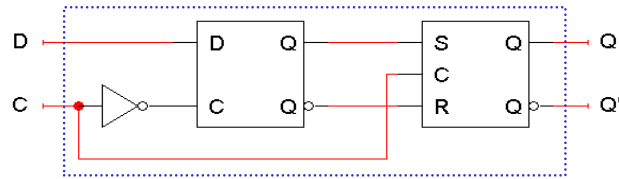
- D flip-flop'un C kontrol girişi ya D latch ini enable yapar yada SR latch i. İkisinin aynı anda enable olması mümkün değil.
- C = 0 ise:
 - master latch enable olur. Ve master latch in çıkışı flip-flop'un D girişini gösterir. Eğer D değişirse master'ın çıkışı da değişir.
 - slave latch disable olur. Dolayısıyla, D latch'in çıkışı onu etkilemez. Bu durumda slave latch in çıkışı dolayısıyla da flip-flop'un çıkışı flip-flop'un şimdiki durumunu gösterir.

Y.Doç.Dr.Tuncay UZUN

6. Ardışıl Lojik Devreler

15

D flip-flop'da C=1 ise,



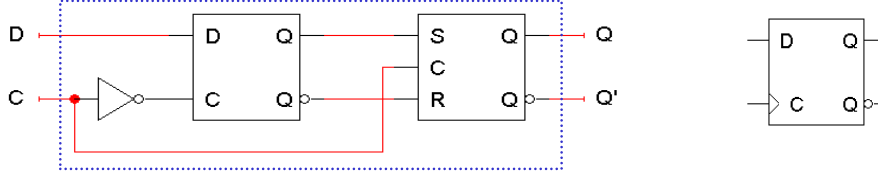
- Ardından C = 1 olurolmaz, *(hatta saatin yükselen kenarında)*
 - Master disable olur. Çıkışı da C=1 olmadan önceki son D giriş değeri olarak kalır.
 - Disable olduğu için D girişinde olacak olan değişiklikler C = 1 olduğu sürece master latchi etkilemez.
 - Slave enable olur. S ve R nin değerine göre slave in çıkışı ve dolayısıyla da flip-flop'un çıkışı belirlenir.

Y.Doç.Dr.Tuncay UZUN

6. Ardışıl Lojik Devreler

16

Pozitif Kenar Tetikleme



- Bu bir **pozitif kenar tetiklemeli** flip-flop'dur. Flip-flop'un Q çıkışı sadece C'nin pozitif kenarının ardından değişebilir.
- D flip-flop'unun davranışı Q'nun pozitif kenarda değişmesi dışında D latch'i ile aynıdır.

C	D	Q
0	x	No change
1	0	0 (reset)
1	1	1 (set)

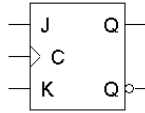
Y.Doç.Dr.Tuncay UZUN

6. Ardışıl Lojik Devreler

17

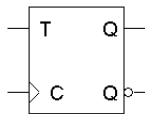
Flip-flop çeşitleri

- D flip-flop'u temel alınarak çeşitli flip-floplar tasarlanmıştır.
- **JK flip-flop** (S ve R a benziyor ama JK=11 flip-flop'un şimdiki durumunun tümleyenini almak için kullanılır.)



C	J	K	Q _{gelecek}
0	x	x	Değişmez
1	0	0	Değişmez
1	0	1	0 (reset)
1	1	0	1 (set)
1	1	1	Q' _{şimdiki}

- **T flip-flop** sadece şimdiki durumu tutar veya tümleyenini alır.



C	T	Q _{gelecek}
0	x	Değişmez
1	0	Değişmez
1	1	Q' _{şimdiki}

Y.Doç.Dr.Tuncay UZUN

6. Ardışıl Lojik Devreler

18

Karakteristik tablolar ve denklemler

- Her bir flip-flop bir **karakteristik tablo** vardır. Bu tablolar giriş ve şimdiki duruma $Q(t)$ bağlı olarak gelecek durumu $Q(t+1)$ gösterirler. (Basitlik açısından, kontrol girişi C tablolara alınmamıştır. Ayrıca, aksi söylenmediği sürece flip-flop'lar pozitif kenar tetiklemelidir.)
- karakteristik denklemler** ise, gelecek durum $Q(t+1)$ 'i şimdiki durum $Q(t)$ ve giriş cinsinden birer fonksiyon ile ifade etmektedirler.

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	$Q'(t)$	Tümlleme

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

T	$Q(t+1)$	İşlem
0	$Q(t)$	Değişmez
1	$Q'(t)$	Tümlleme

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

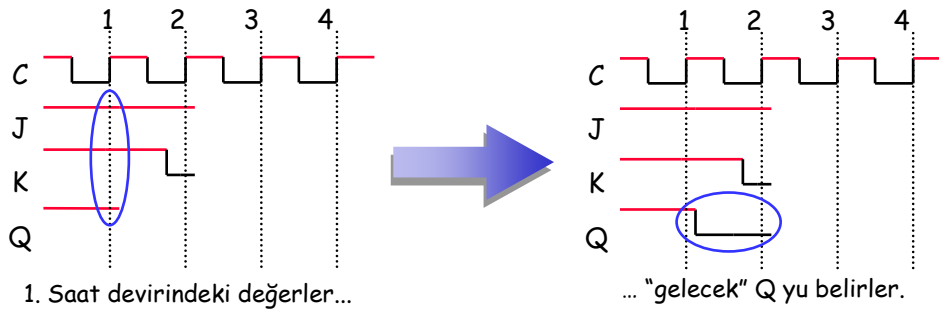
Y.Doç.Dr.Tuncay UZUN

6. Ardışıl Lojik Devreler

19

Flip flop zamanlama diyagramları

- JK flip-flop zamanlama diyagramı örneği: 1. pozitif saat kenarında $J=1$, $K=1$ ve $Q(1) = 1$. O halde gelecek durum $Q(2) = Q(1)' = 0$ oluyor.
- $Q(2)$ 1. pozitif saat kenarının hemen ardından 0 değerini alır. Artık bu değer şimdiki durumdur ve 2. pozitif saat kenarına kadar değişmez.



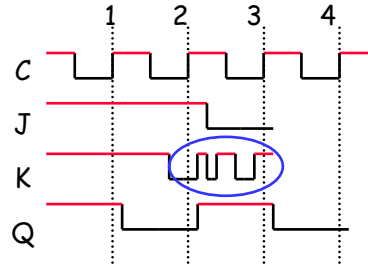
Y.Doç.Dr.Tuncay UZUN

6. Ardışıl Lojik Devreler

20

Önemli Not

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



Y.Doç.Dr.Tuncay UZUN

6. Ardışıl Lojik Devreler

21

6.2.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: $f(\text{şimdiki durum, girişler})$

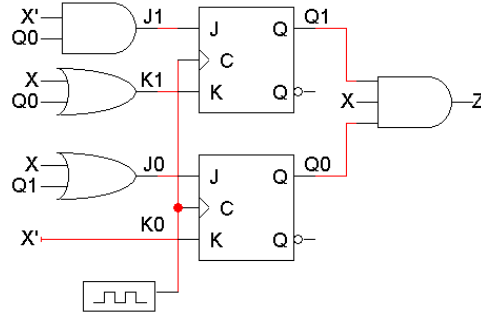


Y.Doç.Dr.Tuncay UZUN

6. Ardışıl Lojik Devreler

22

Bir ardışıl devre örneği



- İki tane JK flip-flop içeren bir ardışıl devre
- **Durum veya Bellek:** Q1Q0
- Bir giriş: X; Bir çıkış: Z

Y.Doç.Dr.Tuncay UZUN

6. Ardışıl Lojik Devreler

23

Bir ardışıl devre nasıl tanımlanır?

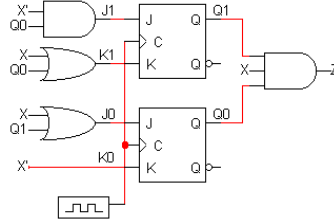
- Kombinezonsal devre – Doğruluk tablosu (Girişler ile çıkışlar arasındaki ilişkiyi gösterir)
- Ardışıl devre – **Durum tablosu** (Girişler ile şimdiki durumların *solda*, çıkışlar ile gelecek durumların ise *sağda* olduğu tablodur.)
 - Şimdiki durumlar ve girişler kullanılarak FF ların gelecek durumlarının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

Y.Doç.Dr.Tuncay UZUN

6. Ardışıl Lojik Devreler

24

Örnek devrenin durum tablosu



Şimdiki Durum		Giriş	Gelecek Durum		Çıkış
Q ₁	Q ₀	X	Q ₁	Q ₀	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			

Y.Doç.Dr.Tuncay UZUN

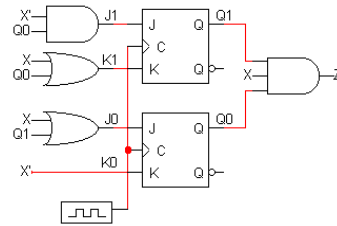
6. Ardışıl Lojik Devreler

25

- Çıkışları yazmak kolay! Devre diyagramından bakılırsa:

$$Z = Q_1 Q_0 X$$

Mealy modeli devresi !!!



Şimdiki Durum		Giriş	Gelecek Durum		Çıkış
Q ₁	Q ₀	X	Q ₁	Q ₀	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

Y.Doç.Dr.Tuncay UZUN

6. Ardışıl Lojik Devreler

26

Flip-flop girişleri

- Gelecek durumları bulmak kolay değildir:

Adım 1:

Flip-flop girişleri için Boole fonksiyon ifadeleri bul.

Adım 2:

Bu fonksiyon ifadelerini kullanarak gerçek flip-flop giriş değerlerini bul (şimdiki durum ve girişler üzerinden olası bütün kombinasyonlar için) **Yani, durum tablosunda arada yeni kolonlar oluştur.**

Adım 3:

Flip-flop karakteristik tablosunu veya denklemlerini kullanarak gelecek durum değerlerini bul (şimdiki durumlar ve girişlere bakarak elde edilmiş olan flip-flop giriş değerlerini kullanarak)

Adım 1: Flip-flop giriş denklemleri

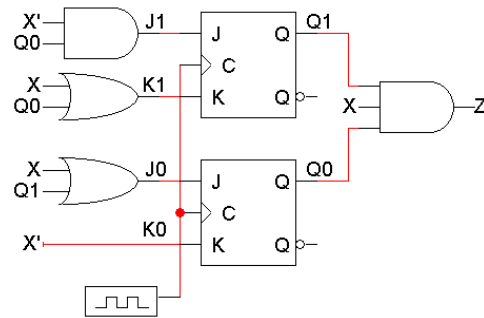
- Bu örnek için, **flip-flop giriş denklemleri:**

$$J_1 = X' Q_0$$

$$K_1 = X + Q_0$$

$$J_0 = X + Q_1$$

$$K_0 = X'$$



Adım 2: Flip-flop giriş değerleri

- Bu denklemleri kullanarak, şimdiki durumlar Q_1Q_0 ve giriş X 'in tüm kombinasyonları için J_1 , K_1 , J_0 ve K_0 'ı hesaplayarak tabloya yaz.

$$J_1 = X' Q_0$$

$$K_1 = X + Q_0$$

$$J_0 = X + Q_1$$

$$K_0 = X'$$

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

Y.Doç.Dr.Tuncay UZUN

6. Ardışıl Lojik Devreler

29

Adım 3: Gelecek durumlar

- Son olarak, JK flip-flop karakteristik tablosu veya denklemlerini yardımıyla, şimdiki durum ve giriş değerlerine bakarak her bir flip-flop'un gelecek durumunu bul
- Genel JK flip-flop'un karakteristik denklemi:

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

- Bu örnekte iki tane JK flip-flop var:

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

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

- 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

Y.Doç.Dr.Tuncay UZUN

6. Ardışıl Lojik Devreler

30

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

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

Şimdiki Durum		Giriş	FF Girişler				Gelecek Durum	
Q_1	Q_0	X	J_1	K_1	J_0	K_0	Q_1	Q_0
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

Y.Doç.Dr.Tuncay UZUN

6. Ardışıl Lojik Devreler

31

Durum tablosunun oluşumu

- Başlangıçta tabloda: **Şimdiki Durum** ve **Girişler**
 - Şimdiki Durum** ve **Girişler** → FF Girişleri
 - Şimdiki Durum** ve FF Girişleri → **Gelecek Durum** (flip-flop karakteristik tablosundan)
 - Şimdiki Durum** ve **Girişler** → **Çıkışlar**

Şimdiki Durum		Giriş	FF Girişleri				Gelecek Durum		Çıkış
Q_1	Q_0	X	J_1	K_1	J_0	K_0	Q_1	Q_0	Z
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

Y.Doç.Dr.Tuncay UZUN

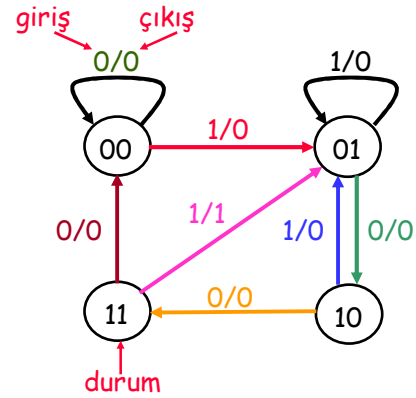
6. Ardışıl Lojik Devreler

32

Durum Diyagramları (Mealy modeli)

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

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



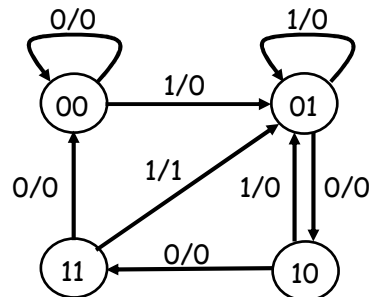
Y.Doç.Dr.Tuncay UZUN

6. Ardışıl Lojik Devreler

33

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.

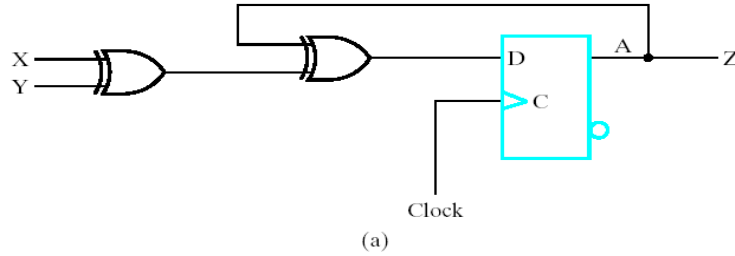


Y.Doç.Dr.Tuncay UZUN

6. Ardışıl Lojik Devreler

34

Moore modeli devresi



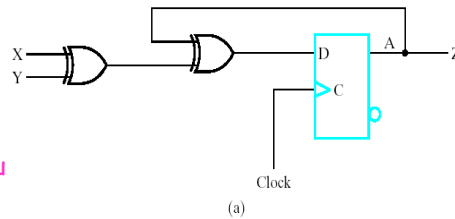
- İki giriş: X ve Y; Bir çıkış: Z
- Bir durum: A
- Dikkat edilirse $Z = A$, yani sadece şimdiki duruma bağlı bir fonksiyon

Y.Doç.Dr.Tuncay UZUN

6. Ardışıl Lojik Devreler

35

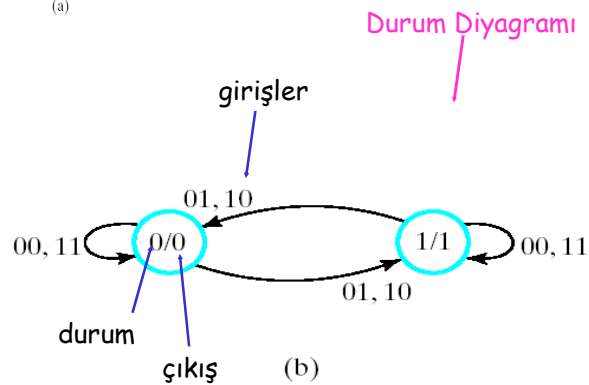
Durum Tablosu ve Durum Diyagramı (Moore modeli)



Durum Tablosu

Present state	Inputs		Next state	Output
A	X	Y	A	Z
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	0
1	0	0	1	1
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

(b) State table



6. Ardışıl Lojik Devreler

36

Ardışıl devre analizi - Özet

- Bir ardışıl devrenin analizini yapmak için:
 - Devrenin çıkışına ve flip-flop'ların girişine ait Boole fonksiyonu ifadelerini bul
 - Bu ifadeleri kullanarak durum tablosundaki çıkış ve flip-flop girişi kolonlarını ilgili değerler ile doldur
 - Son olarak, ilgili flip-flop'a ait karakteristik tabloyu veya denklemleri kullanarak tablodaki gelecek durumkolonlarını doldur.
- Ardışıl devre analizinin sonucu ya bir durum tablosu yada bir durum diyagramı olacaktır.

6.3.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 binary kod atanır (Eğer henüz atanmamışsa). N tane durum varsa, binary 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 excitation tabloları kullanılabilir.

Adım 4:

Flip-flop giriş ve çıkışları için basitleştirilmiş ifadeler bulunur.

Adım 5:

Devre kurulur!

Dizi Yakalayıcısı (Sequence recognizers)

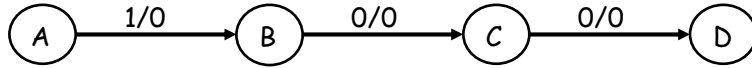
- Bir **dizi yakalayıcısı** devrenin girişinde özel bir bit dizisini arayan özel bir ardışıl devredir.
- Yakalayıcı devrenin tek bir girişi vardır: X.
 - Girişin her bir biti bir saat periyotunda sağlanır.
 - Bu uzun giriş dizilerinin uygulanması için en kolay yoldur.
- Tek bir çıkış vardır: Z. İstenen bit dizisi bulunduğunda 1 çıkışını verir.
- **Örnek:** Aranacak olan bit dizisi: 1001

Inputs: 1 1 1 0 0 1 1 0 1 0 0 1 0 0 1 1 0 ...
 Outputs: 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, 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.

Y.Doç.Dr.Tuncay UZUN 6. Ardışıl Lojik Devreler 41

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


şimdiki durum → giriş/çıkış → gelecek durum

Y.Doç.Dr.Tuncay UZUN 6. Ardışıl Lojik Devreler 42

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	Gelecek Durum		Çıkış
Q_1	Q_0	X	Q_1	Q_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

Y.Doç.Dr.Tuncay UZUN

6. Ardışıl Lojik Devreler

43

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

Y.Doç.Dr.Tuncay UZUN

6. Ardışıl Lojik Devreler

44

Flip-flop giriş değerlerinin bulunması

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

J	K	Q(+1)	Operation
0	0	Q(t)	No change
0	1	0	Reset
1	0	1	Set
1	1	Q'(t)	Complement

- 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.

Y.Doç.Dr.Tuncay UZUN

6. Ardışıl Lojik Devreler

45

JK karakteristik tablosu (excitation table)

- Bir **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(+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

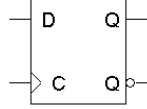
J	K	Q(+1)	İşlem
0	0	Q(t)	Değişmez
0	1	0	Reset
1	0	1	Set
1	1	Q'(t)	Tümleme

Y.Doç.Dr.Tuncay UZUN

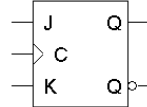
6. Ardışıl Lojik Devreler

46

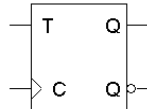
Tüm Flip-Flop(FF)lar için karakteristik tablolar



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	Operation
0	0	0	Değişmez
0	1	1	Tümleyen
1	0	1	Tümleyen
1	1	0	Değişmez

Y.Doç.Dr.Tuncay UZUN

6. Ardışıl Lojik Devreler

47

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 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

Y.Doç.Dr.Tuncay UZUN

6. Ardışıl Lojik Devreler

48

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 don't care durumuna sahip olmalarıdır. Bu sayede daha basit denklemler elde edilir.

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	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

$$J_1 = X' Q_0$$

$$K_1 = X + Q_0$$

$$J_0 = X + Q_1$$

$$K_0 = X'$$

$$Z = Q_1 Q_0 X$$

Y.Doç.Dr. Tuncay UZUN

6. Ardışıl Lojik Devreler

49

Adım 5: Devrenin kurulumu

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

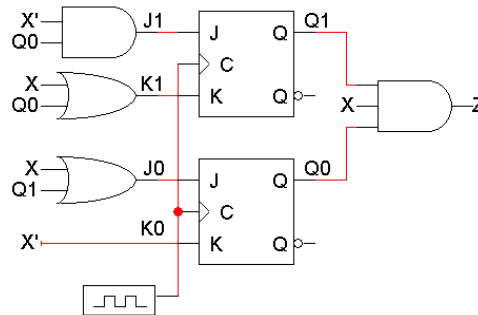
$$J_1 = X' Q_0$$

$$K_1 = X + Q_0$$

$$J_0 = X + Q_1$$

$$K_0 = X'$$

$$Z = Q_1 Q_0 X$$



Y.Doç.Dr. Tuncay UZUN

6. Ardışıl Lojik Devreler

50

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 ₁	Q ₀	X	Q ₁	Q ₀	D ₁	D 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

Y.Doç.Dr.Tuncay UZUN

6. Ardışıl Lojik Devreler

51

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 ₁	Q ₀	X	Q ₁	Q ₀	D ₁	D ₀	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

Y.Doç.Dr.Tuncay UZUN

6. Ardışıl Lojik Devreler

52

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)

