

Lojik Devreler

Latch ve Flip-Flop

Doç. Dr. Fatih Evran
Elektrik-Elektronik Mühendisliği
Ofis:421, Mühendislik Binası
Düzce Üniversitesi
Email: fatihevran@duzce.edu.tr

ARDIŞIL DEVRELER (Sequential Circuits)

Dersin ilk bölümünde **kombinasyonel (combinational) devreleri** inceledik. Bu tür devrelerde çıkışın değeri o andaki girişlerin değerlerine bağlıdır.

Kombinasyonel devre : $\text{Çıkış} = G(\text{Giriş})$ G : çıkış fonksiyonu

- **Ardışıl (sequential) devrelerde** ise çıkış değeri, hem girişlerden gelen değerlere hem de devrenin bir önceki "durumuna" bağlıdır.

Kombinasyonel devre : $\text{Çıkış} = G(\text{Giriş}, \text{Şimdiki Durum})$

Sonraki Durum = $H(\text{Giriş}, \text{Şimdiki Durum})$

Durum bilgisini tutmak için bu devrelerde **bellek** elemanları bulunur. Ardışıl devrelere örnek olarak bozuk parayla çalışan meşrubat makinelerindeki lojik devreler gösterilebilir. Böyle bir sayısal ardışıl devre, ürünü vermek için sadece o anda atılan parayı değil, daha önce atılmış olan parayı da dikkate almalıdır.

Ardışıl devrelerin türleri:

Ardışıl devreler iki gruba ayrılır:

A) Senkron (eş zamanlı) ardışıl devreler:

Bu devreler sadece belli zamanlarda durum değiştirebilirler. Tüm bellek elemanları ortak bir saat işareti ile eş zamanlı (senkron) olarak tetiklenirler.

B) Asenkron ardışıl devreler:

Bu tür devreler her hangi bir zamanda girişlerdeki değişime bağlı olarak durum değiştirebilirler.

Bu derste günümüzde çok yaygın olarak kullanılan eş zamanlı devreler ele alınacaktır. Örneğin mikroişlemciler saatle tetiklenen eş zamanlı ardışıl devrelerdir.

Sonlu Durumlu Makine (Finite State Machine-FSM) Modeli

Ardışıl devreler "sonlu durumlu makine" (Finite State Machine- FSM) modeli kullanılarak tasarlanırlar.

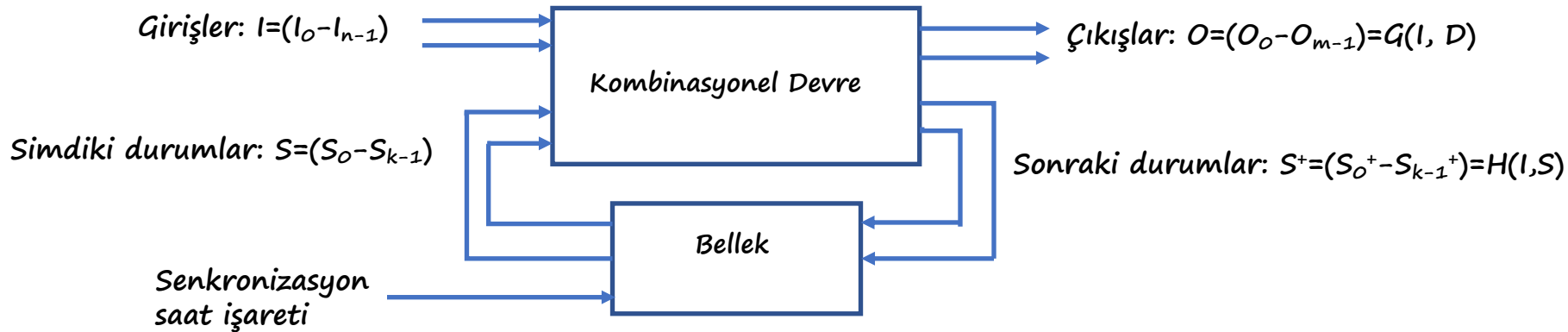
Bu modelleme yöntemi, bir çok başka sistemin tasarımında da kullanılır.

- Böyle bir makine ilk çalışmaya başladığında belli bir durumda bulunur.
- Gelen giriş değerine göre ve içinde bulunduğu duruma göre makine bir çıkış üretir.
- Gelen giriş değerine göre ve içinde bulunduğu duruma göre yeni bir duruma geçer.

Sonlu durumlu makineler, lojik devre olarak olarak gerçekleştirilirken iki kısımdan oluşturulurlar:

a) Lojik işlemleri yapan kombinosyonel devre,

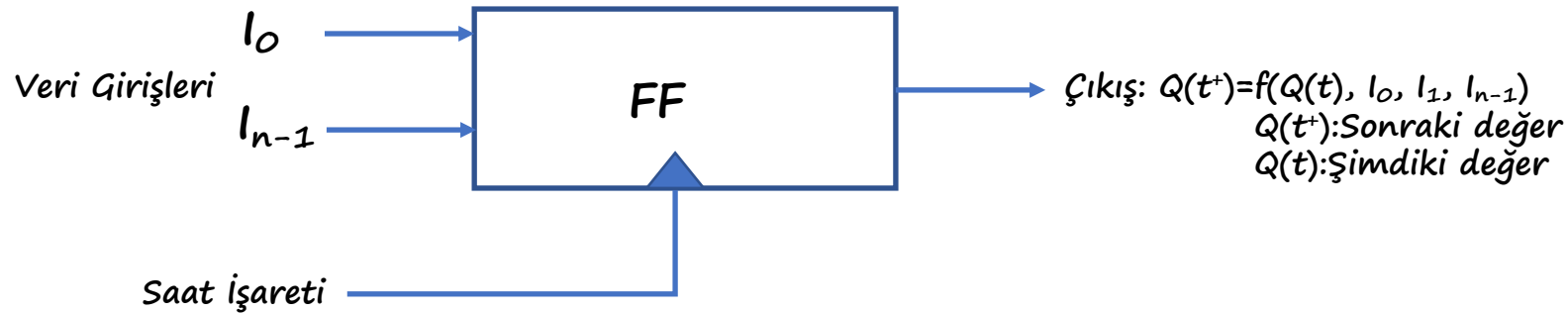
b) Durum bilgisini tutan bellek elemanları.



Bellek elemanları incelendikten sonra ardışıl devreler konusu tekrar ele alınacaktır.

Veri Saklama (Bellek) Elemanları

'Flip-flop': Bir bitlik bellek elemanlarıdır. Çok girişli, bir çıkışlı lojik bir devre olarak tasarlanırlar.



Q çıkışı flip-flopun o anda içindeki ikili değeri (0,1) dışarı yansıtır. Bu değer aynı zamanda flip-flopun durum bilgisidir.

Q çıkışının alacağı yeni değer $Q(t^+)$, veri girişlerinin ve o andaki durumun $Q(t)$ bir fonksiyonu olarak belirlenir.

Saat işareti, veri girişlerindeki değerlerin ne zaman değerlendirileceğini, yani flip-flop'un ne zaman değer değiştireceğini belirten işarettir. Sadece saat işaretinin etkin olduğu anlarda flip-flop'un içeriği yukarıdaki fonksiyona göre belirlenerek değiştirilir.

Saat işareti etkin değilse, veri girişleri değişse bile flip-flop bir önceki içeriğini korur.

Saat (Clock) İşareti:

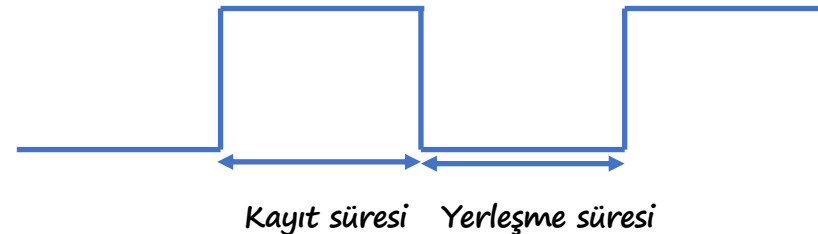
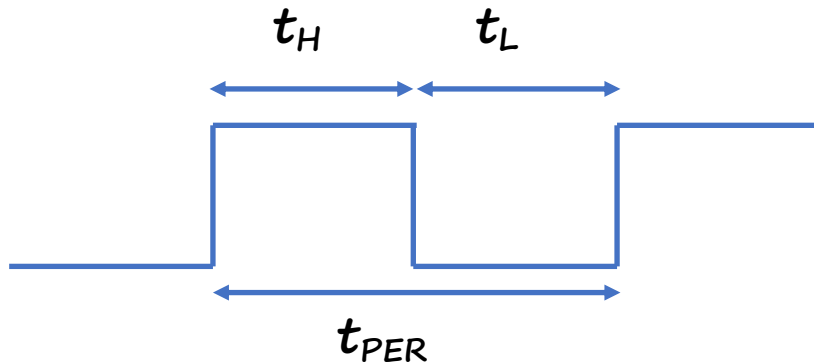
Sayısal sistemlerdeki elemanların eş zamanlı (senkronize) çalışmasını sağlayan dikdörtgen dalga şeklinde bir işarettir. Saat işaretiyle denetlenen elemanlar (örneğin flip-flop) sadece saat işareti etkin olunca işlem yaparlar. Onun dışında eski durumların korurlar.

Saat işaretinin kullanılması açısından elemanlar ikiye ayrılır.

a) Düzey tetiklemeli elemanlar, b) Kenar tetiklemeli elemanlar

Düzey tetiklemeli elemanlar: Saat işaretinin bir düzeyini (pozitif lojikte "1" düzeyini) etkin düzey olarak kabul ederler. Bu elemanlar saat işareti "1" düzeyindeyken işlem yaparak durumlarını ve çıkışlarını değiştirirler; saat işareti "0" düzeyindeyken eski durumlarını korurlar. Saat işaretinin "1" düzeyindeyken girişler işleme sokulduğundan, bu süre boyunca giriş değerleri sabit tutulmalıdır. Aksi durumda ardışıl elemanın çıkışının alacağı değer belirsiz olur. Bu süreye **kayıt süresi** denir.

Saat işaretinin "0" olduğu sürede ise girişler değiştirilebilir. Bu süreye **yerleşme süresi** denir.



Kenar tetiklemeli elemanlar:

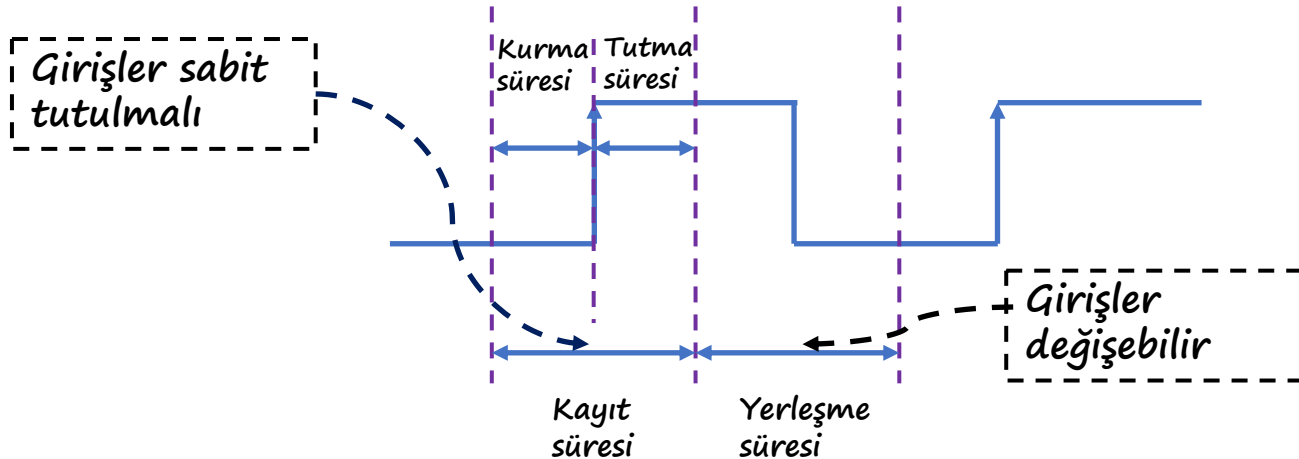
Saat işaretinin bir kenarını (pozitif lojikte çıkan kenar) etkin kenar olarak kabul ederler.

Pozitif kenar tetiklemeli elemanlar saat işareti $0 \rightarrow 1$ geçişi yapınca (çıkan kenar) işlem yaparak durumlarını ve çıkışlarını değiştirirler; saat işareti geçiş yapmazsa eski durumlarını korurlar.

Negatif lojikte ise işlemler $1 \rightarrow 0$ geçişinde (inen kenar) yapılır.

Saat işaretinin $0 \rightarrow 1$ geçişi yaparken girişler işleme sokulduğundan, bu kenardan belli bir süre önce ve sonra giriş değerleri sabit tutulmalıdır. Aksi durumda ardışıl

elemanın çıkışının alacağı değer belirsiz olur.



Kurma süresi: "Set-up time"

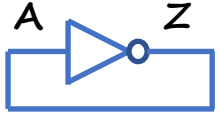
Tutma süresi: "Hold time"

Kayıt süresi, kurma ve tutma sürelerinin toplamından oluşur. Ardışıl devrenin sağlıklı çalışması için bu süre boyunca girişlerin sabit kalması gerekir.

Geri Beslemeli Bağlantılar (Feedback Connections)

Bellek elemanları oluşturabilmek için devrelerde geri besleme bağlantılarına gerek duyulur. Geri beslemenin anlamı; devrenin bazı çıkışlarının aynı devrenin bazı girişlerine bağlanmasıdır.

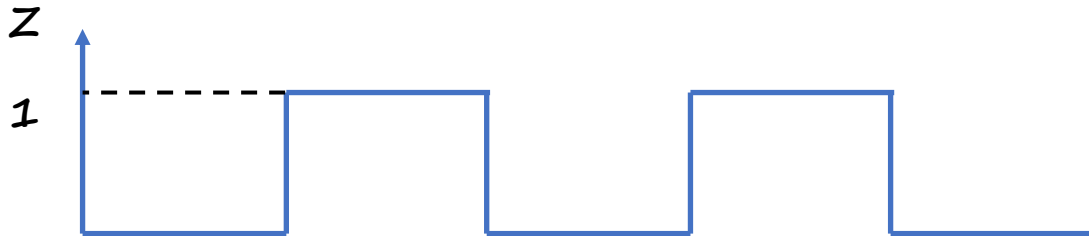
Örnek:



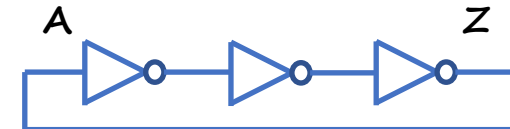
Belli bir anda tümleyicinin A girişi "0" ise bu "0" değeri devrenin Z çıkışının "1" değeri almasına neden olacaktır. Bu "1" değeri tekrar devrenin girişine uygulandığından (geri besleme) yayılma gecikmesi sonrası tümleyicinin Z çıkışı "0" olacaktır. Bu "0" değeri A girişine uygulanacağından Z çıkışı tekrar "1", olacak ve bu değişim bu şekilde devam edecektir.

Aşağıdaki şekilde gösterildiği gibi tümleyicinin Z çıkışı sürekli "0", "1" değerleri arasında değişecek ve hiç bir zaman kararlı bir duruma ulaşmayacaktır. Bu kararsız (unstable) devre bellek olarak kullanılamaz.

Devrenin değişim (osilasyon) periyodu tümleyicinin gecikme süresine bağlıdır.



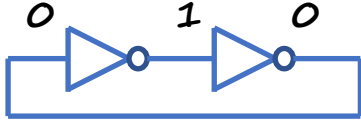
Başka bir kararsız (unstable) devre:



İki Tümlleme Elamanı ile Oluşturulan Geri Beslemeli Devre İki Kararlı Devre (Bistable Circuit)

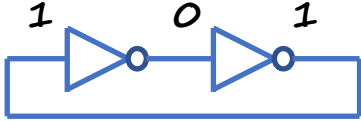
Bu devrede iki tümlleme elemanı kullanılmıştır. Bu durumda devrenin iki kararlı durum olur (bistable).

Kararlı durum 1:



Birinci tümleyicinin girişi "0" ise çıkışı "1" olur. Bu durumda ikinci tümleyicinin girişi "1" olur, çıkışı da "0" olur. Bu "0" değeri ilk tümleyicinin girişine uygulanır (geri besleme). Bu girişte zaten "0" değeri olduğundan devrenin durumunda bir değişiklik olmaz (kararlı). Devre kararlı bir durumdadır.

Kararlı durum 2:



Eğer birinci tümleyicinin girişi "1" ise ikinci tümleyicinin girişi "0" olur ve şekilde gösterildiği gibi ikinci kararlı durum oluşur.

İki Kararlı Durumlu (Bistable) Devre (devamı)

İki kararlı devre yandaki gibi de çizilebilir. Bu devre iki kararlı durumdan birinde bulunur.

Durum 1: $V_{in1} = 1, V_{out1} = V_{in2} = 0, V_{out2} = 1$

Durum 2: $V_{in1} = 0, V_{out1} = V_{in2} = 1, V_{out2} = 0$

Bu elemanın iki kararlı durumu vardır. $Q=0$ ve $Q=1$

Q_N tümleyen çıkıştır: $Q_N = Q'$.

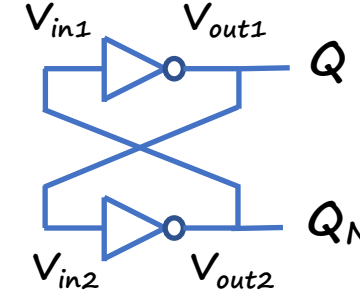
Bunlar belleklerde olması gereken özelliklerdir.

Ancak bu devrenin girişi yoktur. Girişi olmadığından elemanın durumunu dışarıdan denetlemek (değiştirmek) mümkün değildir. İlk gerilim verildiğinde eleman rastgele bir duruma geçer.

Bu nedenle bu devre bellek olarak kullanılamaz.

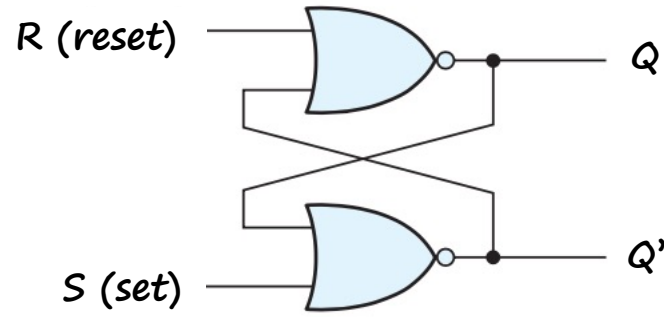
Bellek elemanına aşağıdaki iki özelliğe sahip olması gerekir:

1. İki kararlı durum
2. Durumları değiştirmeyi veya eski durumu korumayı sağlayan denetim girişleri



S-R (Set-Reset) Bilgi Saklama Elemanı (SR Latch-Tutucu)

İki adet iki adet NOR kapısı ile oluşturulabilen bir bitlik saklama elemanıdır. Tüm flip-floplar, bu temel saklama elemanına yapılan eklemeler ile oluşturulabilir.



S: Set (Birleme, Yazma)
R: Reset (Sıfırlama, Silme)
Q: Çıkış (Durum)
Q': Tümlleyen Çıkış (QN)

Kararlı durumda fonksiyon tablosu:

	S	R	Q_n	Q_{n+1}	Q'_n	Q'_{n+1}
SAKLAMA	0	0	0	0	1	1
			1	1	0	0
SİLME	0	1	0	0	1	1
			1	0	0	1
YAZMA	1	0	0	1	1	0
			1	1	0	0
TANIMSIZ	1	1	0	0	1	0
			1	0	0	0

Q_n : Şimdiki durum

Q_{n+1} : Gelecek durum

- S girişi hafıza elemanına "1" yazmak için, R girişi de "0" yazmak için kullanılır.
- Her iki giriş te "0" olduğunda SR elemanı bir önceki durumunu korur.
- Girişlerin her ikisine birden "1" verilmez.

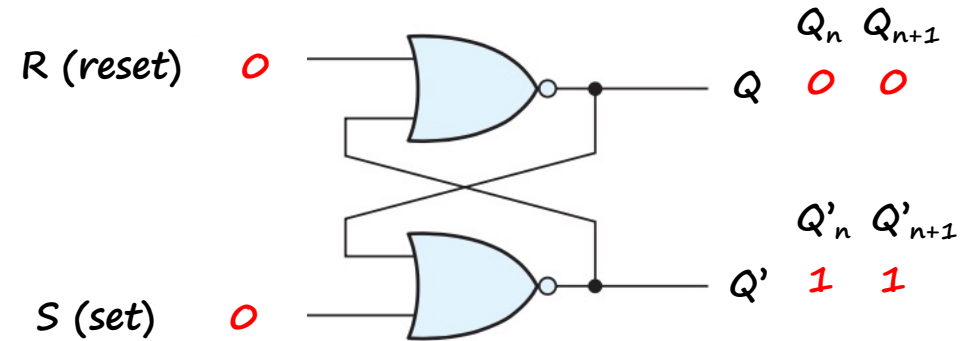
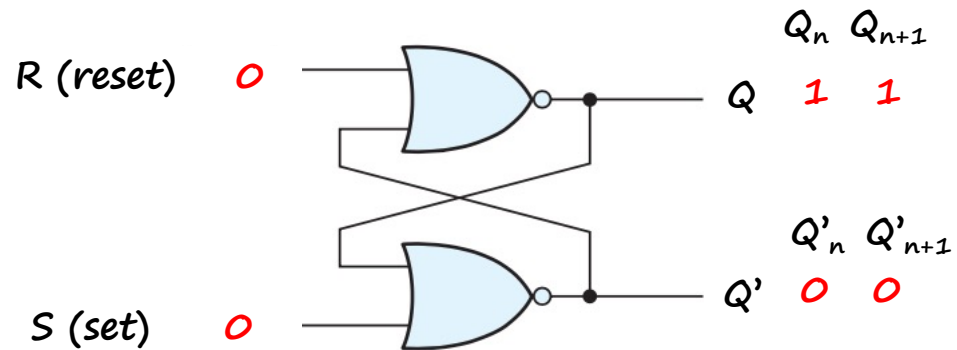
$$Q_{next} = Q_{n+1} = (R + Q'_n)'$$

$$Q'_{next} = Q'_{n+1} = (S + Q_n)'$$

SR Tutucu Analiz (1)

Saklama Durumu:

$SR=00$ ve $Q_n Q'_n=10$ ya da $Q_n Q'_n=01$ olduğunda $Q_{n+1}=(R+Q'_n)'=(0+Q'_n)'=Q_n$ ve $Q'_{n+1}=(S+Q_n)'=(0+Q_n)'=Q'_n$ olmaktadır.



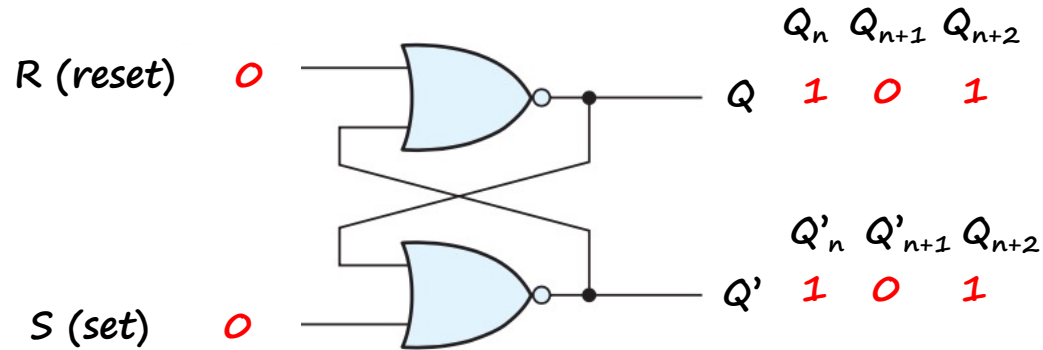
SR Tutucu Analiz (2)

Saklama Durumu:

$SR=00$ ve $Q_n Q'_n=11$ olduğunda

$Q_{n+1}=(R+Q'_n)'=(0+1)'=0$ ve $Q'_{n+1}=(S+Q_n)'=(0+1)'=0$

$Q_{n+2}=(R+Q'_{n+1})'=(0+0)'=1$ ve $Q'_{n+2}=(S+Q_{n+1})'=(0+0)'=1$



Bu durum $SR=11$ 'den $SR=00$ ' a bir değişimin gerçekleşmesinde oluşmaktadır. Çıkışlar kararsızdır yani bilgi saklayamazlar. Kapılardaki gecikme zamanına bağlı olarak çıkışlarda yüksek frekanslı bir dalga şekli oluşmaktadır.

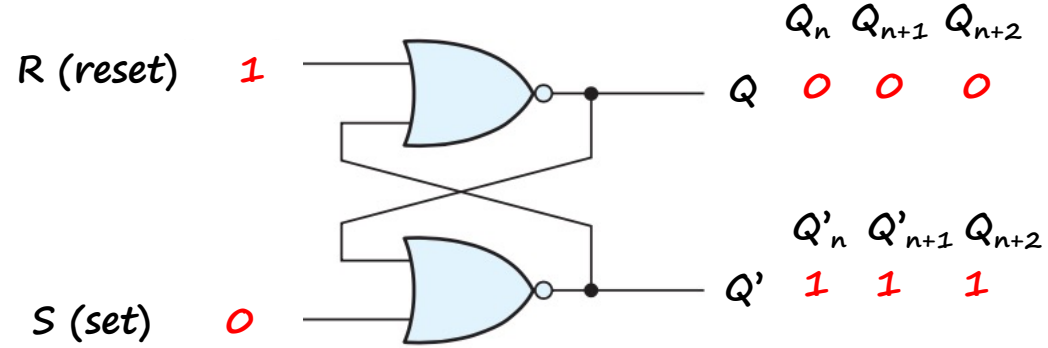
SR Tutucu Analiz (3)

Silme Durumu:

$SR=01$ ve $Q_n Q'_n=01$ olduğunda

$$Q_{n+1} = (R + Q'_n)' = (1 + Q'_n)' = 0$$

$$Q'_{n+1} = (S + Q_n)' = (0 + 0)' = 1$$



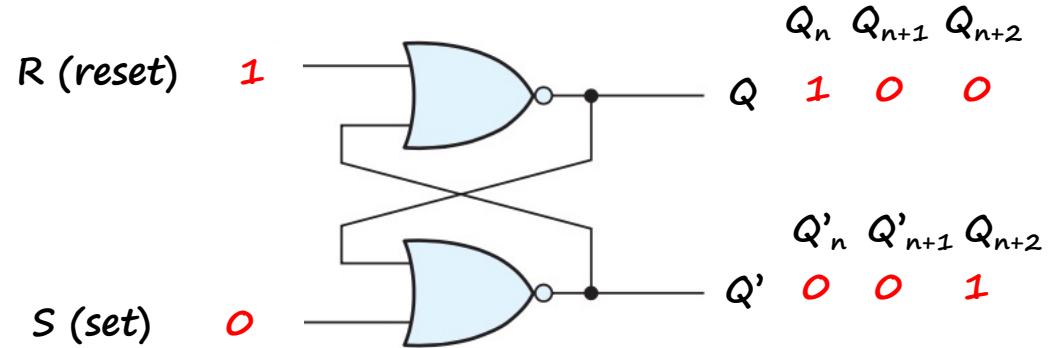
$SR=01$ ve $Q_n Q'_n=10$ olduğunda

$$Q_{n+1} = (R + Q'_n)' = (1 + Q'_n)' = 0$$

$$Q'_{n+1} = (S + Q_n)' = (0 + 1)' = 0$$

$$Q_{n+2} = (R + Q'_{n+1})' = (1 + Q'_{n+1})' = 0$$

$$Q'_{n+2} = (S + Q_{n+1})' = (0 + 0)' = 1$$



Not: Bu işlemde iki adım dolayısıyla da iki birim kapı gecikmesi söz konusu olmasına rağmen devre kararlı olarak çalışmaktadır.

SR Tutucu Analiz (4)

Silme Durumu:

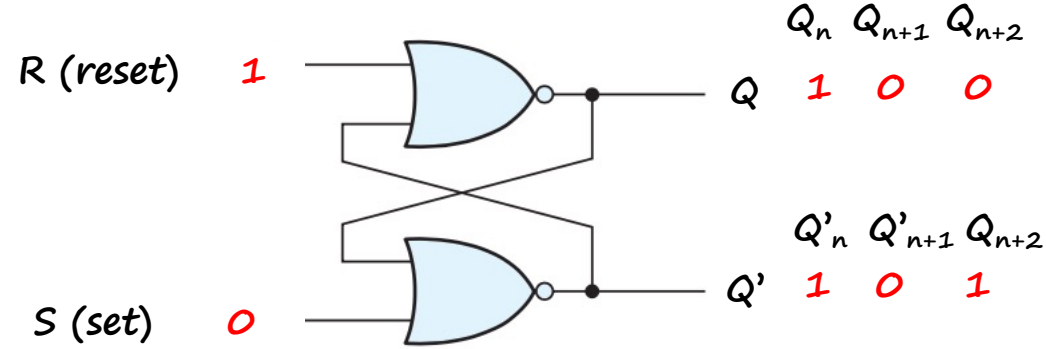
$SR=01$ ve $Q_n Q'_n=11$ olduğunda

$$Q_{n+1}=(R+Q'_n)'=(1+Q'_n)'=0$$

$$Q'_{n+1}=(S+Q_n)'=(0+1)'=0$$

$$Q_{n+2}=(R+Q'_{n+1})'=(1+0)'=0$$

$$Q'_{n+2}=(S+Q_{n+1})'=(0+0)'=1$$

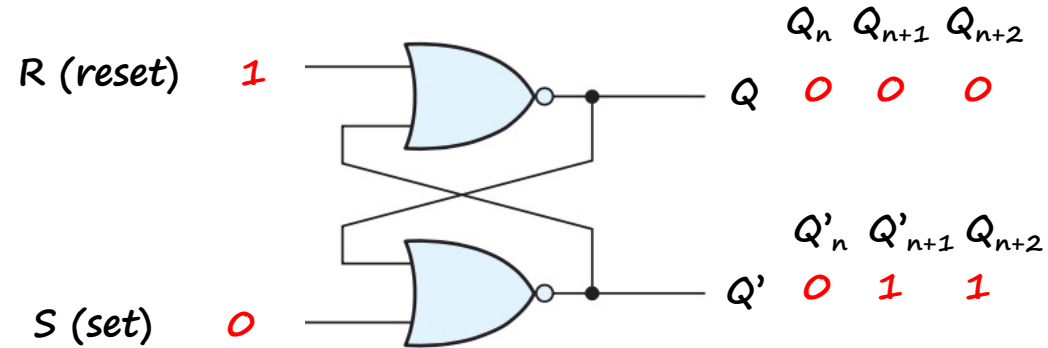


Not: Bu işlemde iki adım dolayısıyla da iki birim kapı gecikmesi söz konusu olmasına rağmen devre kararlı olarak çalışmaktadır.

$SR=01$ ve $Q_n Q'_n=00$ olduğunda

$$Q_{n+1}=(R+Q'_n)'=(1+Q'_n)'=0$$

$$Q'_{n+1}=(S+Q_n)'=(0+0)'=1$$



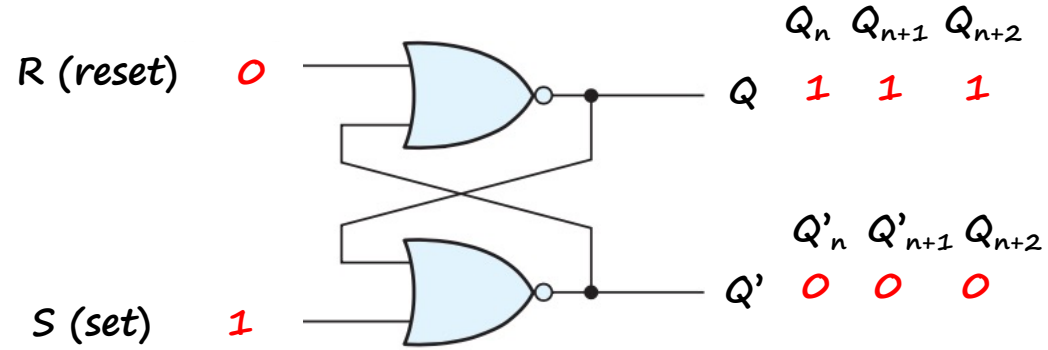
SR Tutucu Analiz (5)

Yazma Durumu:

$SR=10$ ve $Q_n Q'_n=10$ olduğunda

$$Q_{n+1} = (R + Q'_n)' = (0 + Q'_n)' = Q_n = 1$$

$$Q'_{n+1} = (S + Q_n)' = (1 + Q_n)' = 0$$



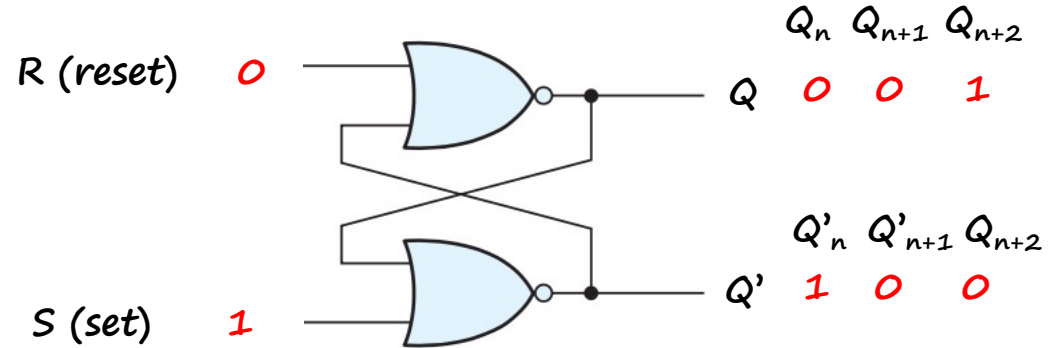
$SR=10$ ve $Q_n Q'_n=01$ olduğunda

$$Q_{n+1} = (R + Q'_n)' = (0 + Q'_n)' = Q_n = 0$$

$$Q'_{n+1} = (S + Q_n)' = (1 + Q_n)' = 0$$

$$Q_{n+2} = (R + Q'_{n+1})' = (0 + 0)' = 1$$

$$Q'_{n+2} = (S + Q_{n+1})' = (1 + 0)' = 0$$



Not: Bu işlemde iki adım dolayısıyla da iki birim kapı gecikmesi söz konusu olmasına rağmen devre kararlı olarak çalışmaktadır.

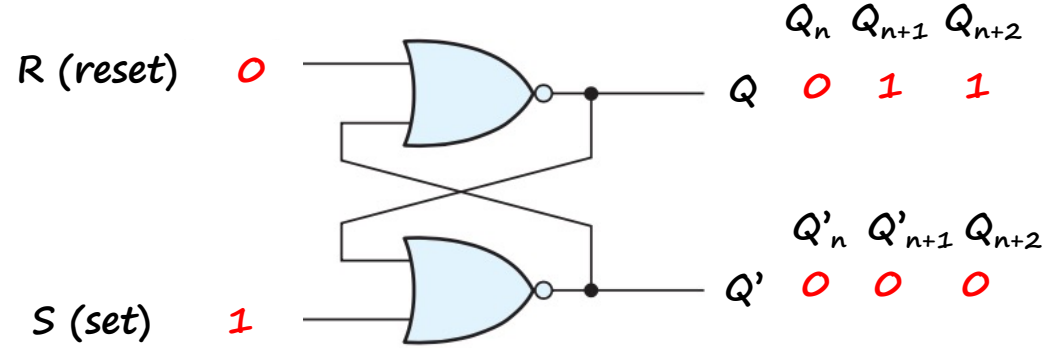
SR Tutucu Analiz (6)

Yazma Durumu:

$SR=10$ ve $Q_n Q'_n=00$ olduğunda

$$Q_{n+1} = (R + Q'_n)' = (0 + 0)' = 1$$

$$Q'_{n+1} = (S + Q_n)' = (1 + 0)' = 0$$



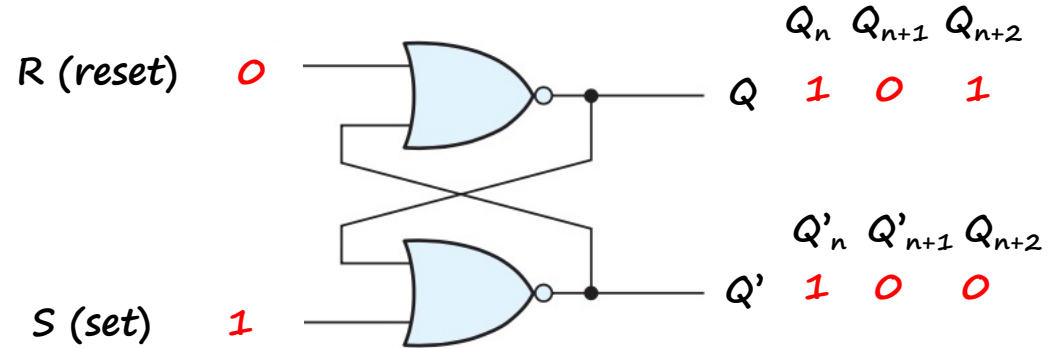
$SR=10$ ve $Q_n Q'_n=11$ olduğunda

$$Q_{n+1} = (R + Q'_n)' = (0 + 1)' = 0$$

$$Q'_{n+1} = (S + Q_n)' = (1 + 1)' = 0$$

$$Q_{n+2} = (R + Q'_{n+1})' = (0 + 0)' = 1$$

$$Q'_{n+2} = (S + Q_{n+1})' = (1 + 0)' = 0$$



Not: Bu işlemde iki adım dolayısıyla da iki birim kapı gecikmesi söz konusu olmasına rağmen devre kararlı olarak çalışmaktadır.

SR Tutucu Analiz (7)

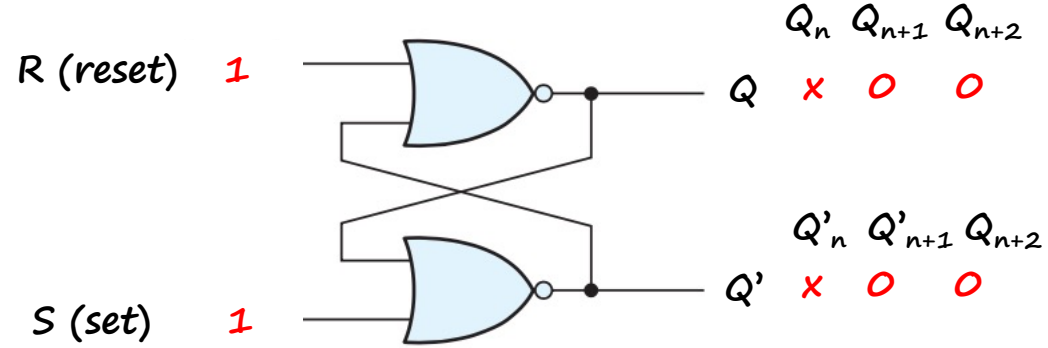
Tanımsız Durumu:

$SR=11$ ve $Q_n Q'_n = xx$ olduğunda

$$Q_{n+1} = (R + Q'_n)' = (1 + x)' = 0$$

$$Q'_{n+1} = (S + Q_n)' = (1 + x)' = 0$$

Olmaktadır.



Not: Çıkışların her ikisi de '0' olmaktadır. $SR=11$ 'den $SR=00$ ' a bir değişim olduğunda ise çıkışlar kararsızdır.

S-R Tutucunun Doğruluk Tablosu ve Karakteristik Denklemi:

Q çıkışının bir sonraki değeri (sonraki durumu) $Q(n+1)$, girişlere ve saklama elemanının o anki çıkışına (durumuna) $Q(n)$ bağlıdır.

Doğruluk Tablosu

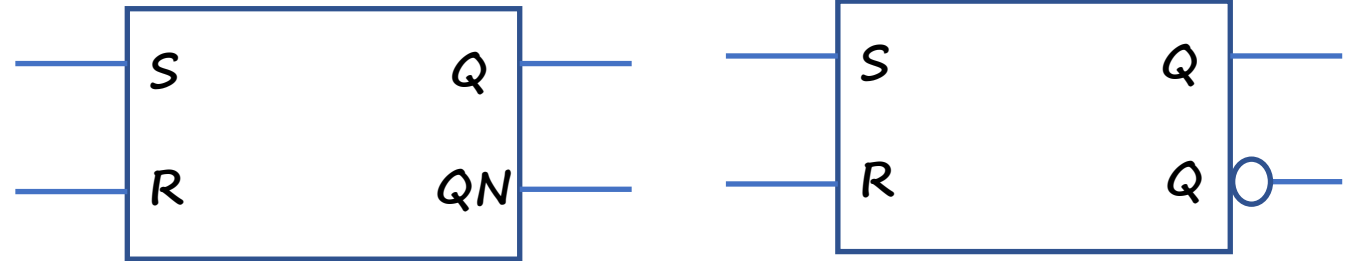
S	R	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	Tanımsız (X)
1	1	1	Tanımsız (X)

S	RQ_n	$\overline{Q_n}$		Q_n		$\overline{Q_n}$	
		00	01	11	10	10	00
0	0	0	1	0	0	0	0
1	1	1	1	X	X	X	X

Karakteristik denklem:

$$Q_{n+1} = S + R'Q_n, \quad SR = 0$$

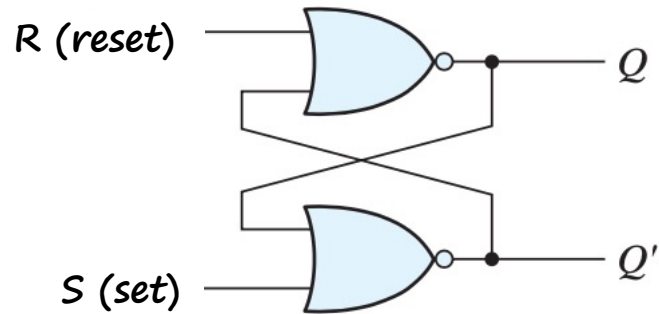
S-R Tutucunun Blok Diyagramları:



S-R Tutucu-Özet

Karakteristik tablosu:

Tanım	S	R	Q(n)	Q(n+1)
SAKLAMA	0	0	0	0
TUTMA	0	0	1	1
SİLME	0	1	0	0
SİLME	0	1	1	0
YAZMA	1	0	0	1
YAZMA	1	0	1	1
TANIMSIZ	1	1	0	Yasak
TANIMSIZ	1	1	1	Yasak



Uyarma tablosu:

Tanım	Q(n)	Q(n+1)	S	R
SAKLAMA/SİLME	0	0	0	d
YAZMA	0	1	1	0
SİLME	1	0	0	1
SAKLAMA/YAZMA	1	1	d	0

Doğruluk tablosu:

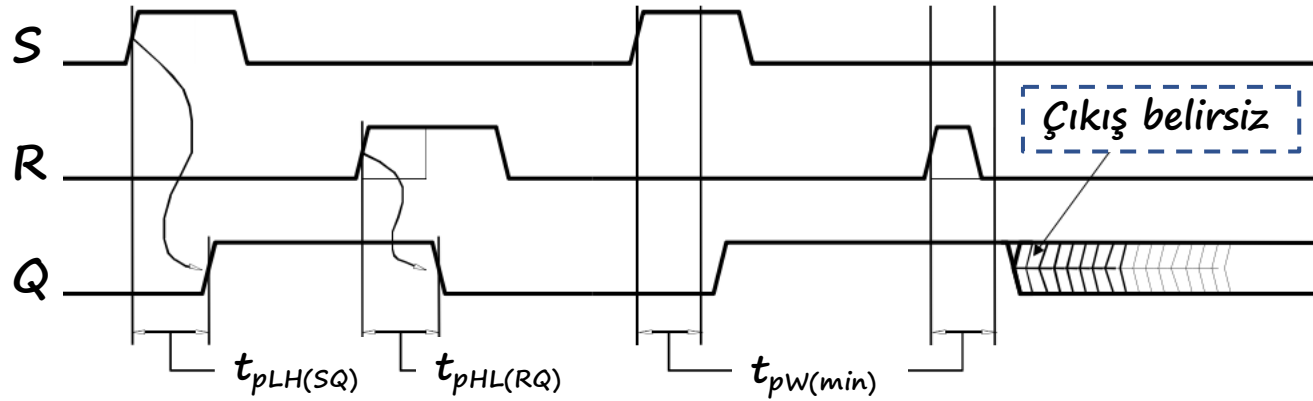
Tanım	S	R	Q(n+1)
SAKLAMA	0	0	Q(n)
SİLME	0	1	0
YAZMA	1	0	1
TANIMSIZ	1	1	Yasak

S-R Tutucunun Zamanlama Özellikleri (1)

Elemanın içindeki yayılma (propagasyon) gecikmesinden dolayı S veya R girişlerindeki değişimlerin etkisi belli bir süre geçtikten sonra çıkışta etkili olur.

Bu süre boyunca girişler sabit kalmalıdır. Aksi durumda çıkışın alacağı değer belirsiz olur.

S (veya R) girişindeki darbenin uzunluğu en az gecikme kadar olmalıdır.



$t_{pLH(SQ)}$: S değiştiğinde çıkışın 0-1 değişim yapması için geçen süre.

$t_{pHL(RQ)}$: R değiştiğinde çıkışın 1-0 değişim yapması için geçen süre.

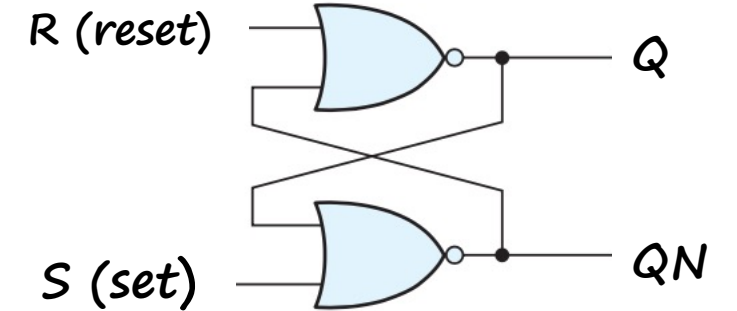
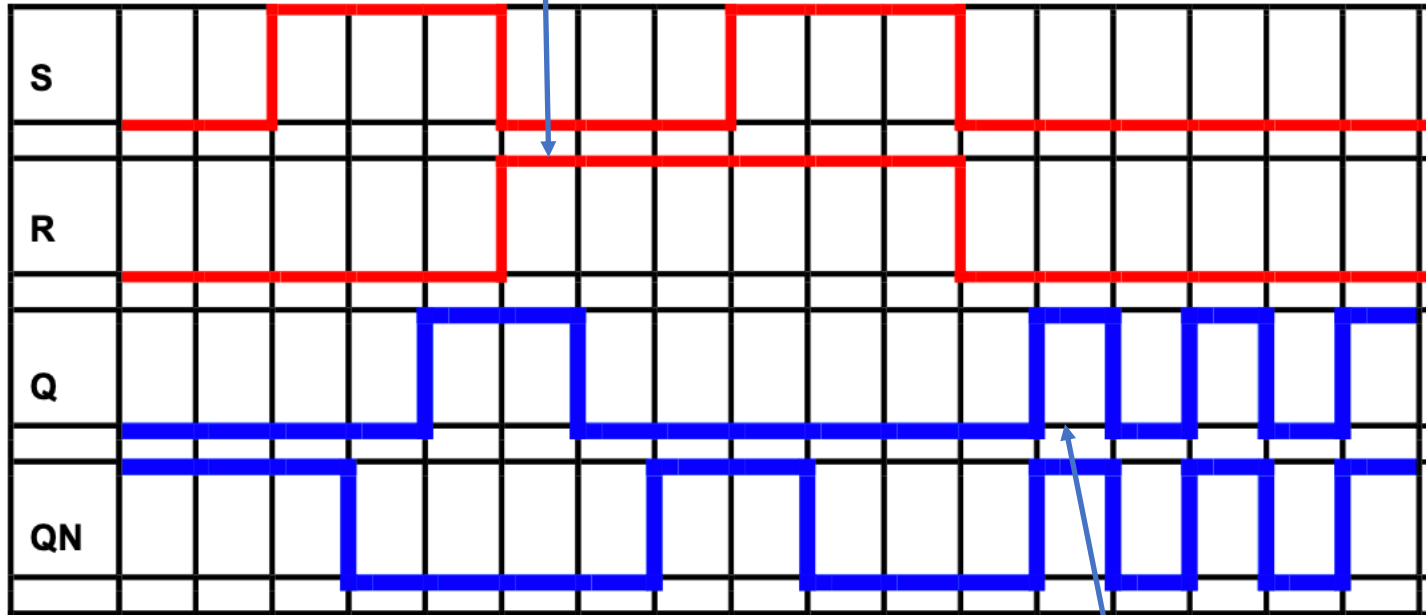
$t_{pW(min)}$: Girişlerin sabit kalması gereken en küçük süre.

S-R Tutucunun Zamanlama Özellikleri (2)

Örnek: Kapılardaki gecikmeyi eşit ve bir kare alınız

SR=10 olduğunda Q, 2 kapı gecikmesinden sonra 1' e ve QN, bir kapı gecikmesinden sonra 0'a değişir.

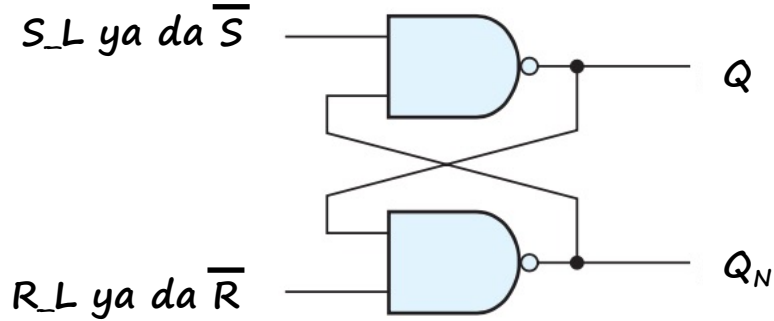
SR=01 olduğunda Q, 1 kapı gecikmesinden sonra 0 ve QN, 2 kapı gecikmesinden sonra 1 olur.



Belirsiz çıkış

NAND Kapılı S-R Tutucu ($\bar{S}$ - $\bar{R}$ Tutucu)

NOR yerine NAND kapıları kullanılarak bir S-R tutucu gerçekleştirilebilir. Elde edilen tutucu $\bar{S}$ - $\bar{R}$ tutucu olarak adlandırılır.



$\bar{S}$: Set (birleme) tümleyeni
R: Reset (sıfırlama) tümleyeni
Q: Çıkış (durum)
Q_N: Tümleyen çıkış ($\bar{Q}$)

Hatırlatma:

- Bir NAND kapısının girişlerinden biri "0" ise, diğer giriş ne olursa olsun çıkış "1" olacaktır.
- Çıkış, yalnızca her iki giriş te "1" olduğunda "0" olur.

Karakteristik tablosu:

S_L	R_L	Q(n)	Q(n+1)	Q _N (n)	Q _N (n+1)
1	1	0	0	1	1
1	0	0	0	1	1
0	1	0	1	1	0
0	0	0	1	1	1

Kapılardaki yayılma gecikmesi tabloda dikkate alınmamıştır.

Q(n): Şimdiki durum

Q(n+1): Sonraki (gelecek) durum

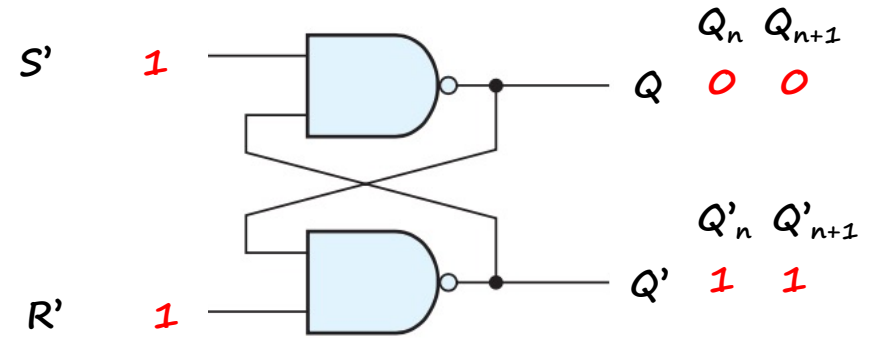
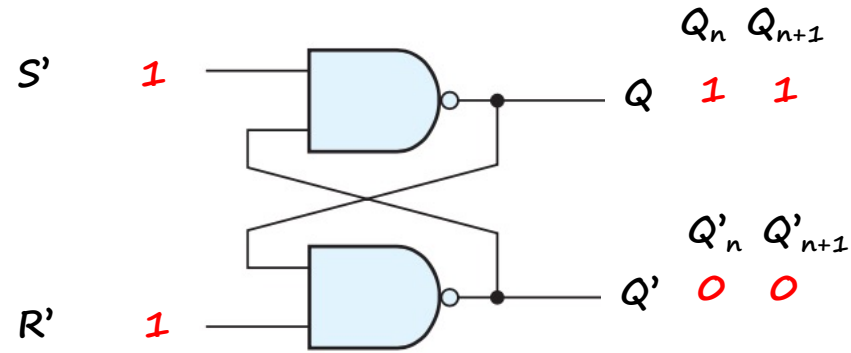
$$Q(\text{next}) = Q(n+1) = (S_L \cdot Q_N(n))'$$

$$Q_N(\text{next}) = Q_N(n+1) = (R_L \cdot Q(n))'$$

S'R' Tutucu Analiz (1)

Saklama Durumu:

$S'R'=11$ ve $Q_nQ'_n=10$ ya da $Q_nQ'_n=01$ olduğunda $Q_{n+1}=(S'.Q'_n)'=(1.Q'_n)'=Q_n$ ve $Q'_{n+1}=(R'.Q_n)'=(1.Q_n)'=Q'_n$ olmaktadır.



$$Q_{next} = Q_{n+1} = (S'.Q'_n)'$$
$$Q'_{next} = Q'_{n+1} = (R'.Q_n)'$$

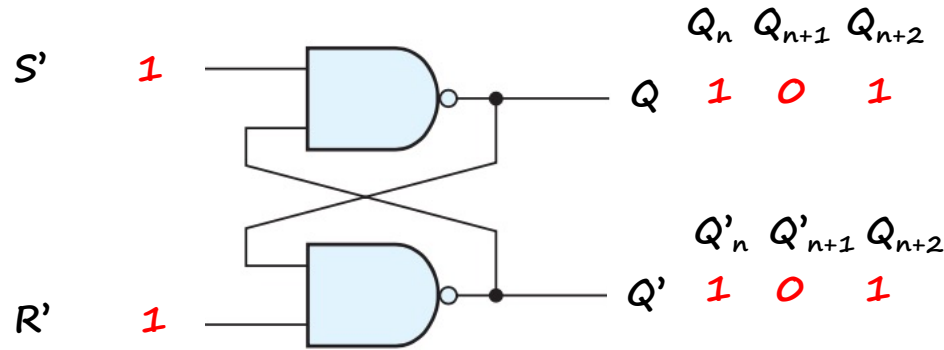
S'R' Tutucu Analiz (2)

Saklama Durumu:

$S'R'=11$ ve $Q_nQ'_n=11$ olduğunda

$$Q_{n+1}=(S'.Q'_n)'=(1.1)'=0, Q'_{n+1}=(R'.Q_n)'=(1.1)'=0$$

$$Q_{n+2}=(S'.Q'_{n+1})'=(1.0)'=1, Q'_{n+2}=(R'.Q_{n+1})'=(1.0)'=1$$



$$Q_{next}=Q_{n+1}=(S'.Q'_n)'$$
$$Q'_{next}=Q'_{n+1}=(R'.Q_n)'$$

Bu durum $S'R'=00'$ dan $S'R'=11'$ e bir değişimin gerçekleşmesinde oluşmaktadır. Çıkışlar kararsızdır yani bilgi saklayamazlar. Kapılardaki gecikme zamanına bağlı olarak çıkışlarda yüksek frekanslı bir dalga şekli oluşmaktadır.

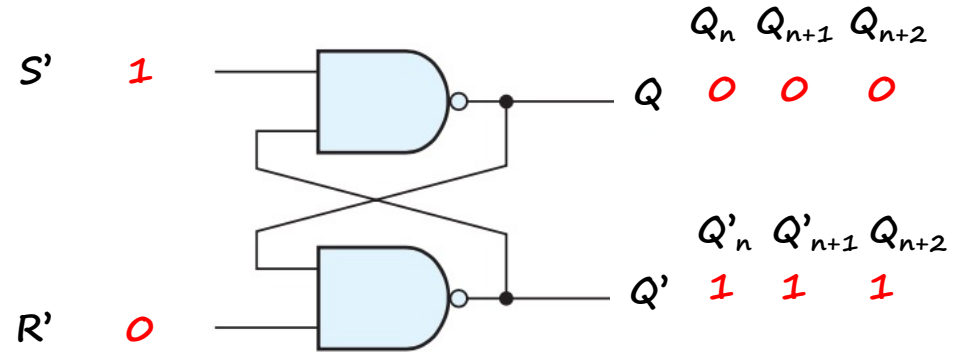
S'R' Tutucu Analiz (3)

Silme Durumu:

$$S'R'=10, Q_n Q'_n=01$$

$$Q_{n+1}=(S'.Q'_n)'=(Q'_n)'=0$$

$$Q'_{n+1}=(R'.Q_n)'=(0.1)'=1$$



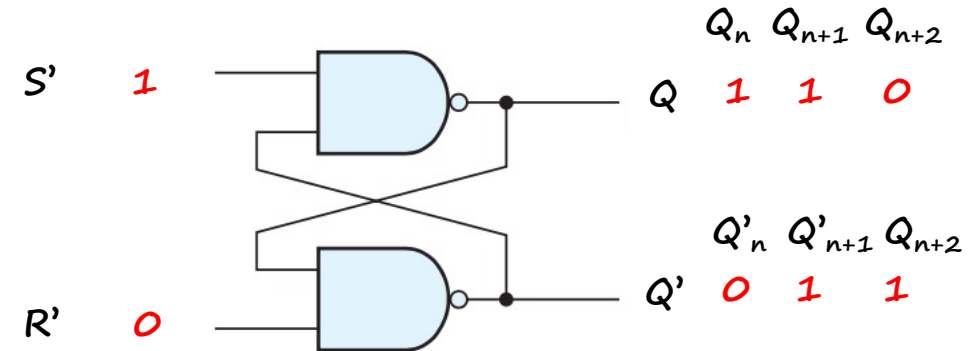
$$S'R'=10, Q_n Q'_n=10$$

$$Q_{n+1}=(S'.Q'_n)'=(1.0)'=1$$

$$Q'_{n+1}=(R'.Q_n)'=(0.1)'=1$$

$$Q_{n+2}=(S'.Q'_{n+1})'=(1.1)'=0$$

$$Q'_{n+2}=(R'.Q_{n+1})'=(0.1)'=1$$



Not: Bu işlemde iki adım dolayısıyla da iki birim kapı gecikmesi söz konusu olmasına rağmen devre kararlı olarak çalışmaktadır.

S'R' Tutucu Analiz (4)

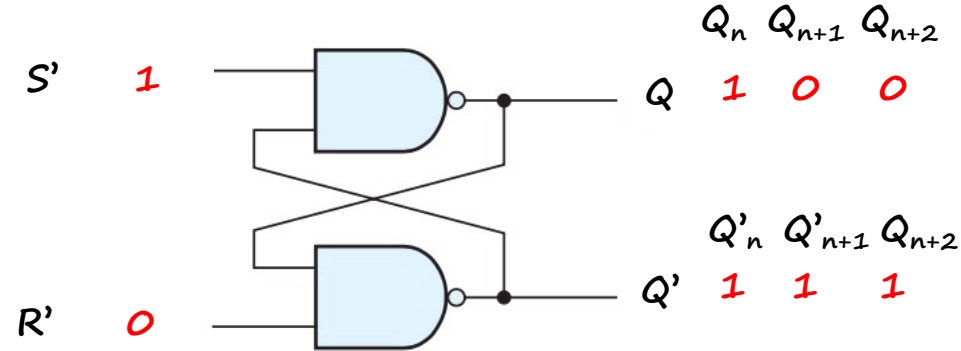
Silme Durumu:

$$S'R'=10, Q_nQ'_n=11$$

$$Q_{n+1}=(S'.Q'_n)'=(1.1)'=0$$

$$Q'_{n+1}=(R'.Q_n)'=(0.1)'=1$$

$$Q_{next}=Q_{n+1}=(S'.Q'_n)'$$
$$Q'_{next}=Q'_{n+1}=(R'.Q_n)'$$



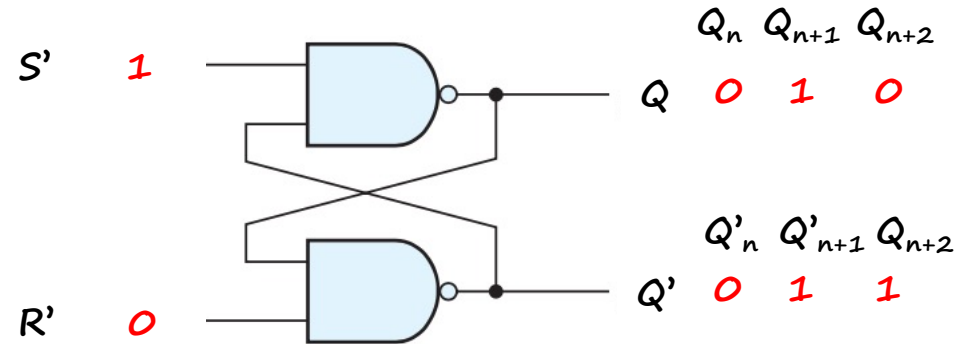
$$S'R'=10, Q_nQ'_n=00$$

$$Q_{n+1}=(S'.Q'_n)'=(1.0)'=1$$

$$Q'_{n+1}=(R'.Q_n)'=(0.0)'=1$$

$$Q_{n+2}=(S'.Q'_{n+1})'=(1.1)'=0$$

$$Q'_{n+1}=(R'.Q_{n+1})'=(0.1)'=1$$



Not: Bu işlemde iki adım dolayısıyla da iki birim kapı gecikmesi söz konusu olmasına rağmen devre kararlı olarak çalışmaktadır.

S'R' Tutucu Analiz (5)

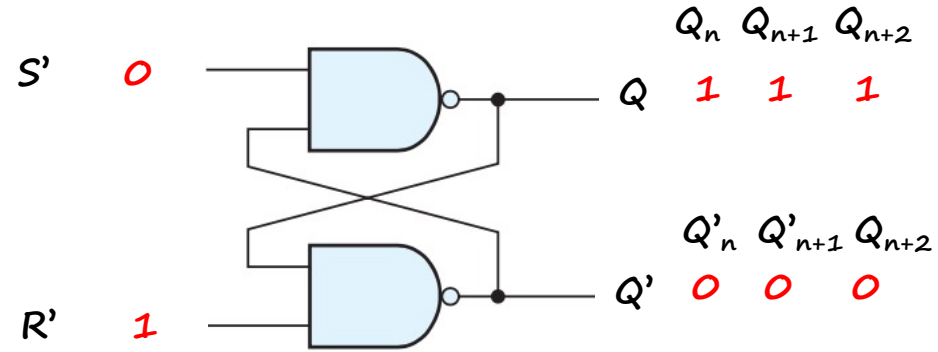
$$Q_{next}=Q_{n+1}=(S'.Q'_n)'$$
$$Q'_{next}=Q'_{n+1}=(R'.Q_n)'$$

Yazma Durumu:

$$S'R'=01, Q_nQ'_n=10$$

$$Q_{n+1}=(S'.Q'_n)'=(1.Q'_n)'=Q_n=1$$

$$Q'_{n+1}=(R'.Q_n)'=(1.Q_n)'=Q'_n=0$$



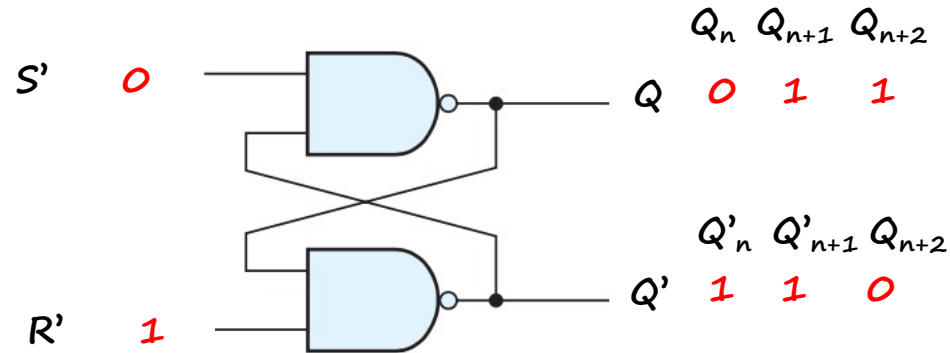
$$S'R'=01, Q_nQ'_n=01$$

$$Q_{n+1}=(S'.Q'_n)'=(0.1)'=1$$

$$Q'_{n+1}=(R'.Q_n)'=(1.0)'=1$$

$$Q_{n+2}=(S'.Q'_{n+1})'=(0.1)'=1$$

$$Q'_{n+2}=(R'.Q_{n+1})'=(1.1)'=0$$



Not: Bu işlemde iki adım dolayısıyla da iki birim kapı gecikmesi söz konusu olmasına rağmen devre kararlı olarak çalışmaktadır.

S'R' Tutucu Analiz (6)

$$Q_{next}=Q_{n+1}=(S'.Q'_n)'$$
$$Q'_{next}=Q'_{n+1}=(R'.Q_n)'$$

Yazma Durumu:

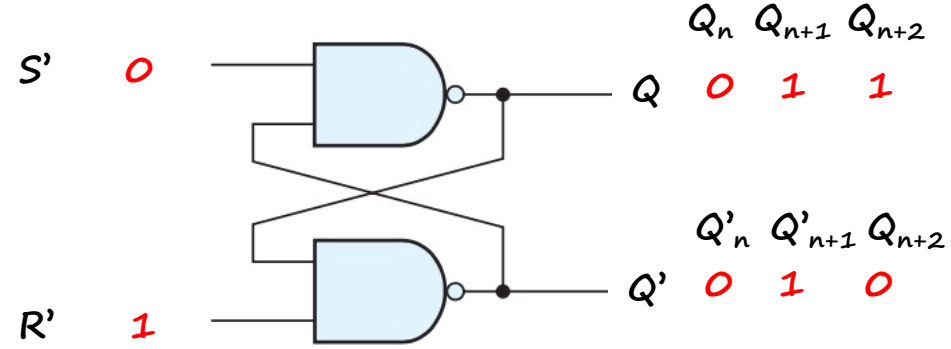
$S'R'=01$ ve $Q_nQ'_n=00$ olduğunda

$$Q_{n+1}=(S'.Q'_n)'=(0.0)'=1$$

$$Q'_{n+1}=(R'.Q_n)'=(1.0)'=1$$

$$Q_{n+2}=(S'.Q'_{n+1})'=(0.1)'=1$$

$$Q'_{n+2}=(R'.Q_{n+1})'=(1.1)'=0$$

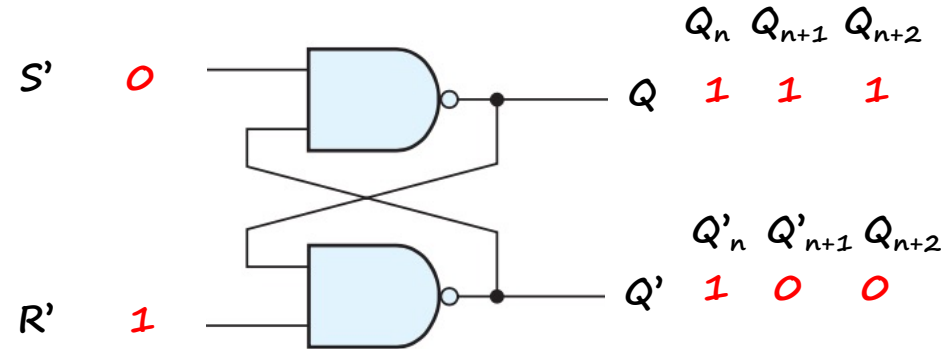


Not: Bu işlemde iki adım dolayısıyla da iki birim kapı gecikmesi söz konusu olmasına rağmen devre kararlı olarak çalışmaktadır.

$S'R'=01$ ve $Q_nQ'_n=11$ olduğunda

$$Q_{n+1}=(S'.Q'_n)'=(0.1)'=1$$

$$Q'_{n+1}=(R'.Q_n)'=(1.1)'=0$$



S'R' Tutucu Analiz (7)

$$Q_{next}=Q_{n+1}=(S'.Q'_n)'$$
$$Q'_{next}=Q'_{n+1}=(R'.Q_n)'$$

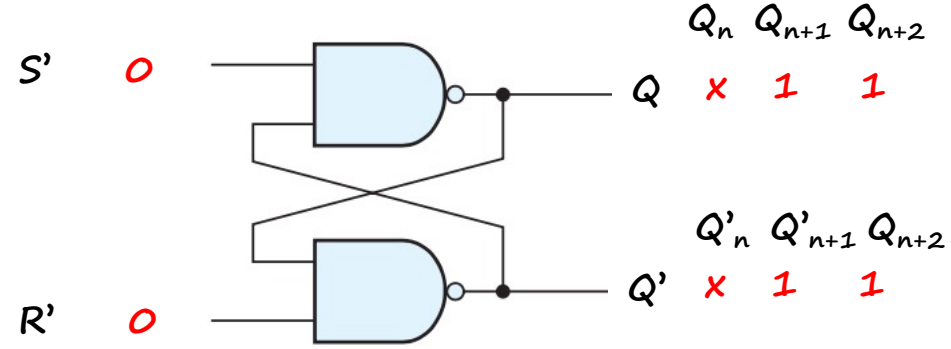
Tanımsız Durumu:

$S'R'=00$ ve $Q_nQ'_n=xx$ olduğunda

$$Q_{n+1}=(S'.Q'_n)'=(0.x)' = 1$$

$$Q'_{n+1}=(R'.Q_n)'=(0.x)' = 1$$

Olmaktadır.



Not: Çıkışların her ikisi de '1' olmaktadır. $S'R'=00$ dan $S'R'=11$ e bir geçiş olduğunda ise çıkışlar kararsız duruma geçmektedir.

S'-R' Tutucunun Karakteristik Denklem ve Tablosu:

Çıkışın sonraki değeri (sonraki durum) $Q(n+1)$, girişlere ve mevcut çıkışa (durum) $Q(n)$ bağlıdır.

Karakteristik tablosu:

S_L	R_L	Q(n)	Q(n+1)
0	0	0	Yasak
0	0	1	Yasak
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

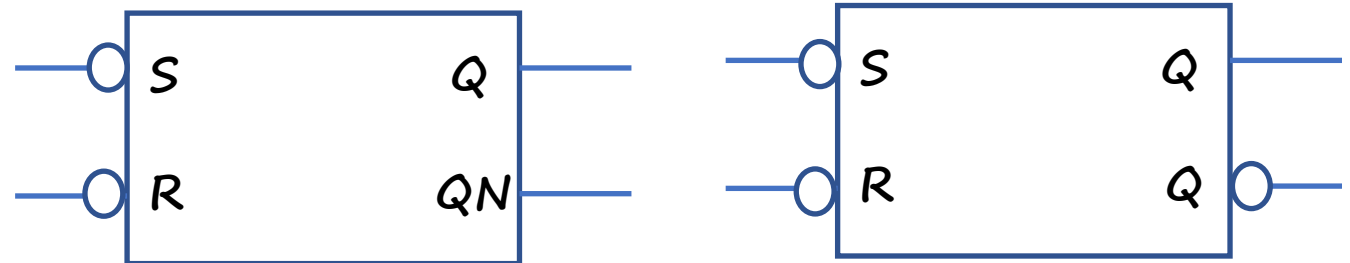
S_L		$R_L Q_n \overline{Q(n)}$		$Q(n)$		$\overline{Q(n)}$	
		00	01	11	10		
$\overline{S_L}$	0	0x	1x	31	21		
	1	40	50	71	60		
		$\overline{R_L}$			R_L		

Karakteristik denklem:
 $Q(n+1) = \overline{S_L} + R_L \cdot Q(n)$

$S_L + R_L = 1$

$S_L = R_L = 0$ (yasaklanmış giriş)

S'-R' Tutucunun Blok Diyagramları:

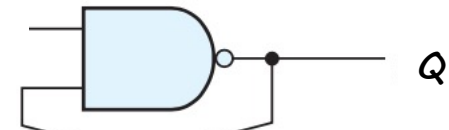


$\overline{S}$ - $\overline{R}$ Tutucu-Özet

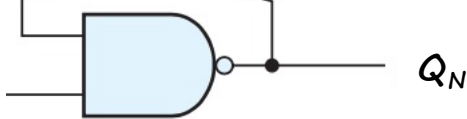
Karakteristik tablosu:

Tanım	S_L	R_L	Q(n)	Q(n+1)
TANIMSIZ	0	0	0	Yasak
	0	0	1	Yasak
YAZMA	0	1	0	1
	0	1	1	1
SİLME	1	0	0	0
	1	0	1	0
TUTMA	1	1	0	0
	1	1	1	1

S_L ya da $\overline{S}$



R_L ya da $\overline{R}$



Doğruluk tablosu:

Tanım	S_L	R_L	Q(n+1)
TANIMSIZ	0	0	Yasak
YAZMA	0	1	1
SİLME	1	0	0
TUTMA	1	1	Q(n)

Uyarma tablosu:

Tanım	Q(n)	Q(n+1)	S_L	R_L
TUTMA/SİLME	0	0	1	d
YAZMA	0	1	0	1
SİLME	1	0	1	0
TUTMA/YAZMA	1	1	d	1

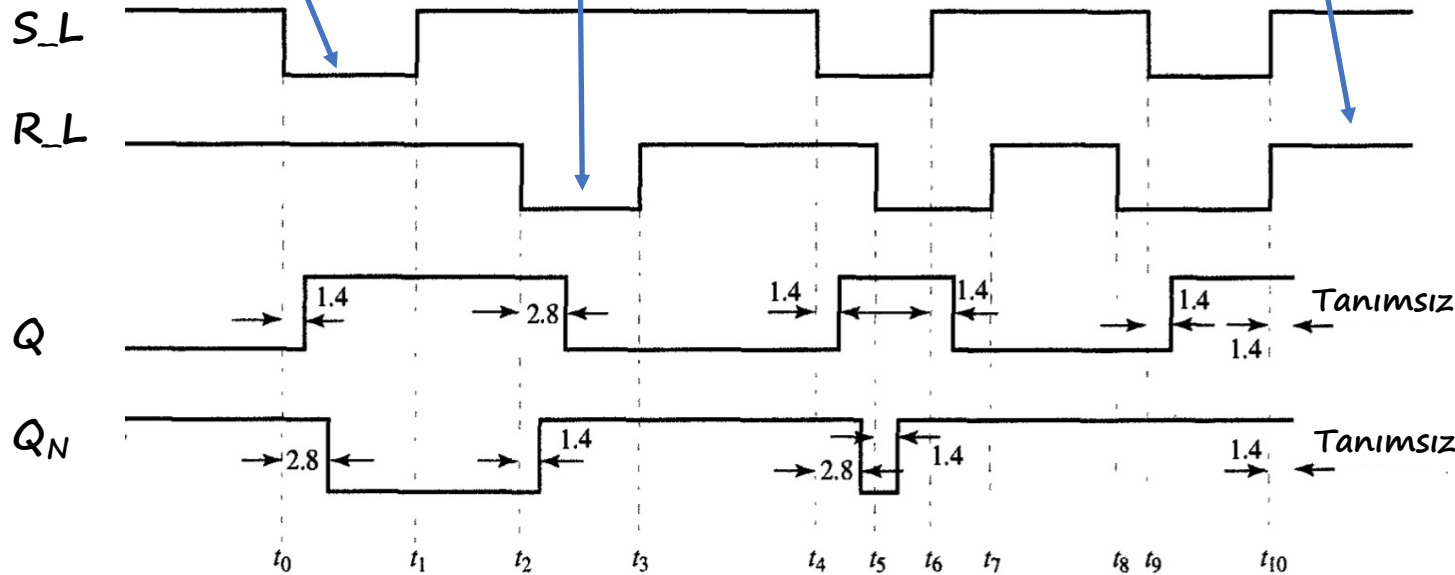
$\bar{S}$ - $\bar{R}$ Tutucunun Zamanlama Özellikleri

$S'R'$ tutucu için SR tutucuna benzer bir analiz yapıldığında aşağıda gösterilen devrenin zamanlama diyagramı elde edilebilir. Her kapının yayılma gecikmesi 1,4 ns' dir.

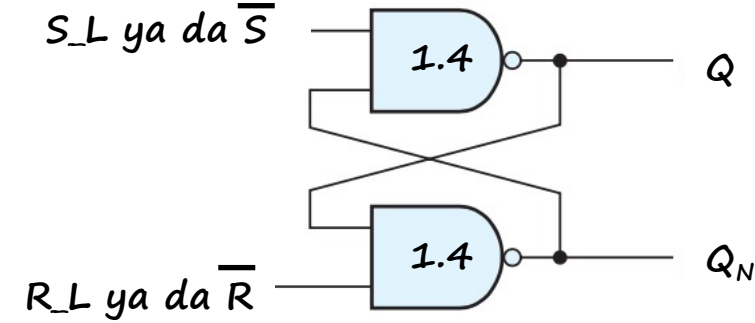
$S_L R_L = 01$ olduğunda, bir kapı gecikmesinden sonra Q 1' e ve iki kapı gecikmesinden sonra Q_N 0' a değişir.

$S_L R_L = 10$ olduğunda, iki kapı gecikmesinden sonra Q 0' a ve bir kapı gecikmesinden sonra Q_N 1' e değişir.

$S_L R_L = 00$ olduğunda, $Q = Q_N$ nedeniyle çıkış belirsizdir

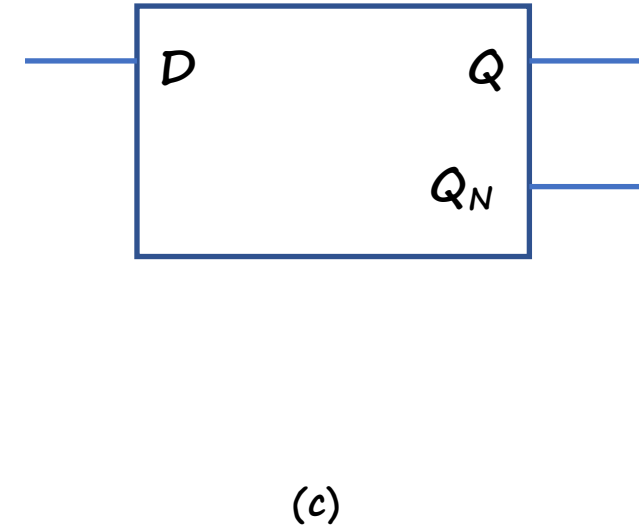
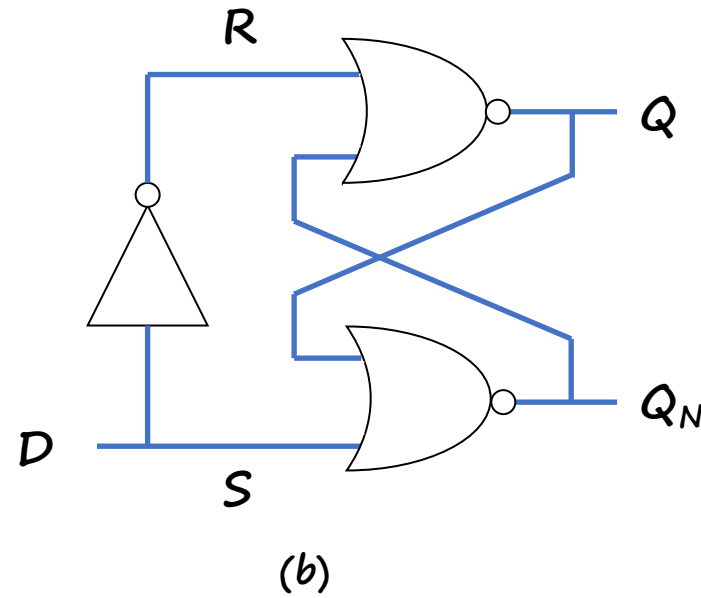
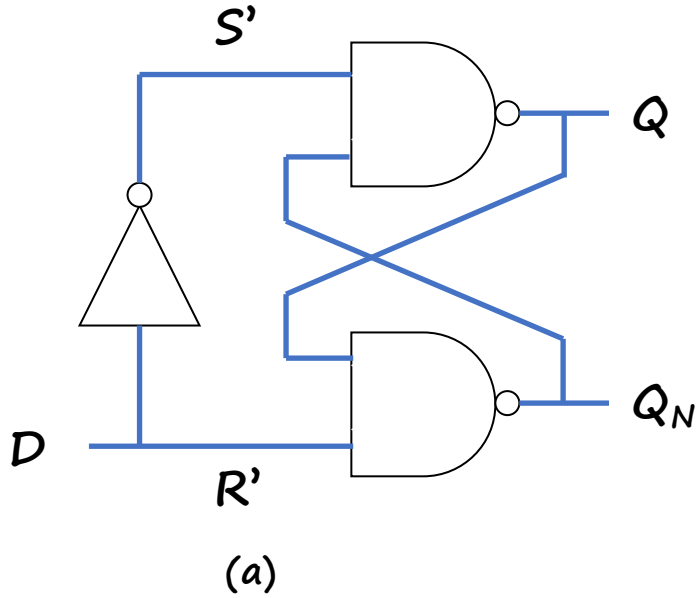


Zamanlama diyagramı



D Tutucu (Transparan)

S-R tutucunun yapısına bazı eklemeler yaparak değişik fonksiyonlara sahip başka tipte tutucular elde edilebilir. S-R tutucunun S ve R girişleri bir tümleyen kapısı ile birleştirilirse D tipi tutucu elde edilir. Bu tutucunun yalnızca bir girişi vardır: D ('data' veya 'delay' anlamına gelir). Transparan, girişteki herhangi bir değişikliğin çıkışta görülebileceği anlamına gelir.



D tutucu: (a) NAND kapıları ile kullanan devre; (b) NAND kapıları ile kullanan devre; (c) blok diyagramı.

D Tutucu (devam)

Karakteristik tablosu:

Tanım	D	Q(n)	Q(n+1)
SİLME	0	0	0
	0	1	0
YAZMA	1	0	1
	1	1	1

Doğruluk tablosu:

Tanım	D	Q(n+1)
SİLME	0	0
YAZMA	1	1

“Belirsiz (tanımsız)” durumlar olmadığına dikkat edin!

Karnaugh haritasının yardımıyla karakteristik tablodan D tutucu:

Uyarma tablosu:

Tanım	Q(n)	Q(n+1)	D
SİLME	0	0	0
YAZMA	0	1	1
SİLME	1	0	0
YAZMA	1	1	1

Q(n)	D		D	
	$\bar{D}$		D	
$\bar{Q}(n)$	0	0	2	1
Q(n)	1	0	3	1

Karakteristik denklem:

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

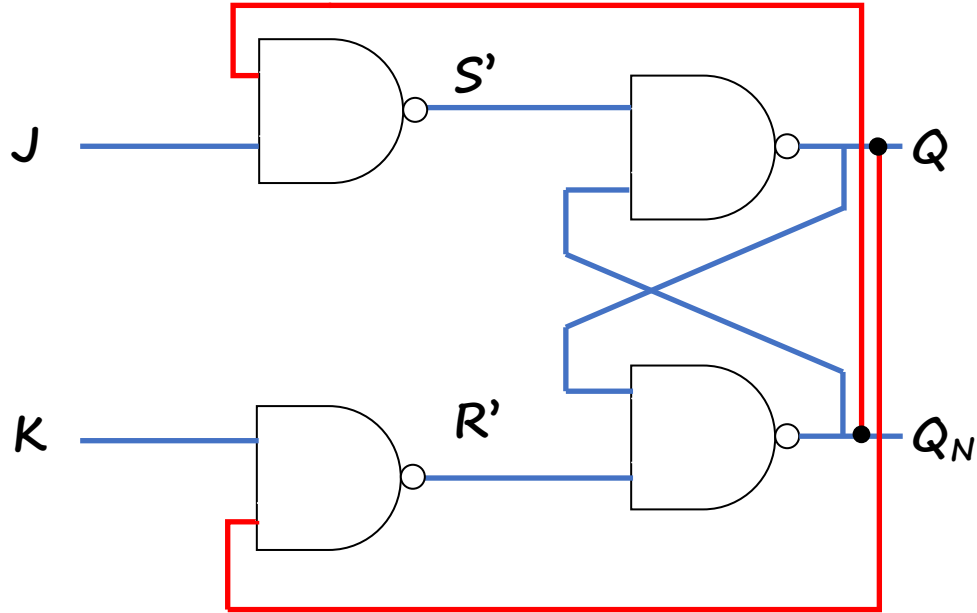
JK Tutucu

SR flip-floplarındaki yasaklı giriş ($S=1, R=1$) problemi JK tipi saklama elemanları ile çözülmüştür. Bu elemanlar SR elemanları gibi çalışır.

J girişi birleme (yazma), K girişi ise sıfırlama (silme) işlemi yapar.

$J=1, K=1$ girişi uygulanması durumunda elemanın içeriği tümlenmiş olur.

Entegre elektronığın doğuşuna yaptığı katkılardan dolayı Jack Kilby' nin onuruna JK adı buradan gelmektedir



(a)



(b)

JK Tutucu: (a) NAND kapılarını kullanan devre; (b) blok diyagram.

JK Tutucu (devam)

Karakteristik tablosu:

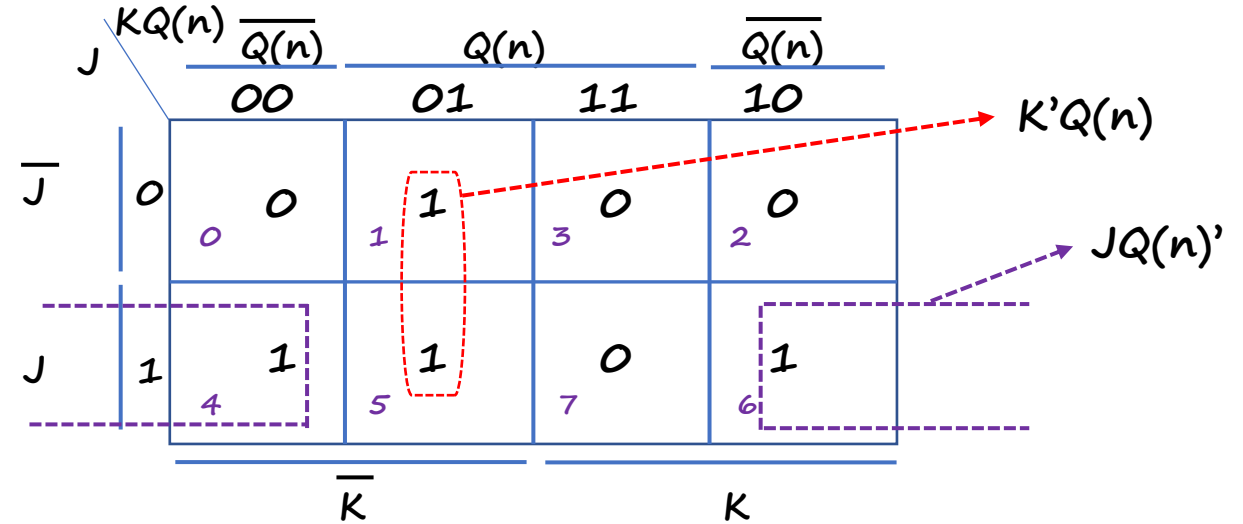
Tanım	J	K	Q(n)	Q(n+1)
SAKLAMA	0	0	0	0
	0	0	1	1
SİLME	0	1	0	0
	0	1	1	0
YAZMA	1	0	0	1
	1	0	1	1
TÜMLEME	1	1	0	1
	1	1	1	0

Uyarma tablosu:

Tanım	Q(n)	Q(n+1)	J	K
SİLME/TUTMA	0	0	0	d
YAZMA/TÜMLEME	0	1	1	d
SİLME/TÜMLEME	1	0	d	1
YAZMA/TUTMA	1	1	d	0

Doğruluk tablosu

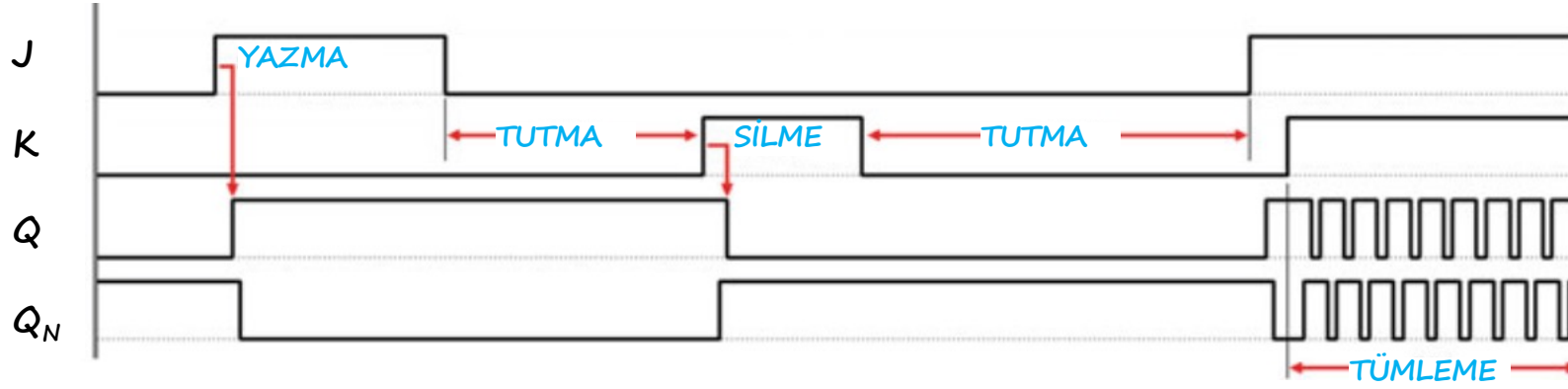
Tanım	J	K	Q(n+1)
SAKLAMA	0	0	Q(n)
SİLME	0	1	0
YAZMA	1	0	1
TÜMLEME	1	1	Q(n)'



Karakteristik denklem:

$$Q(n+1) = JQ(n)' + K'Q(n)$$

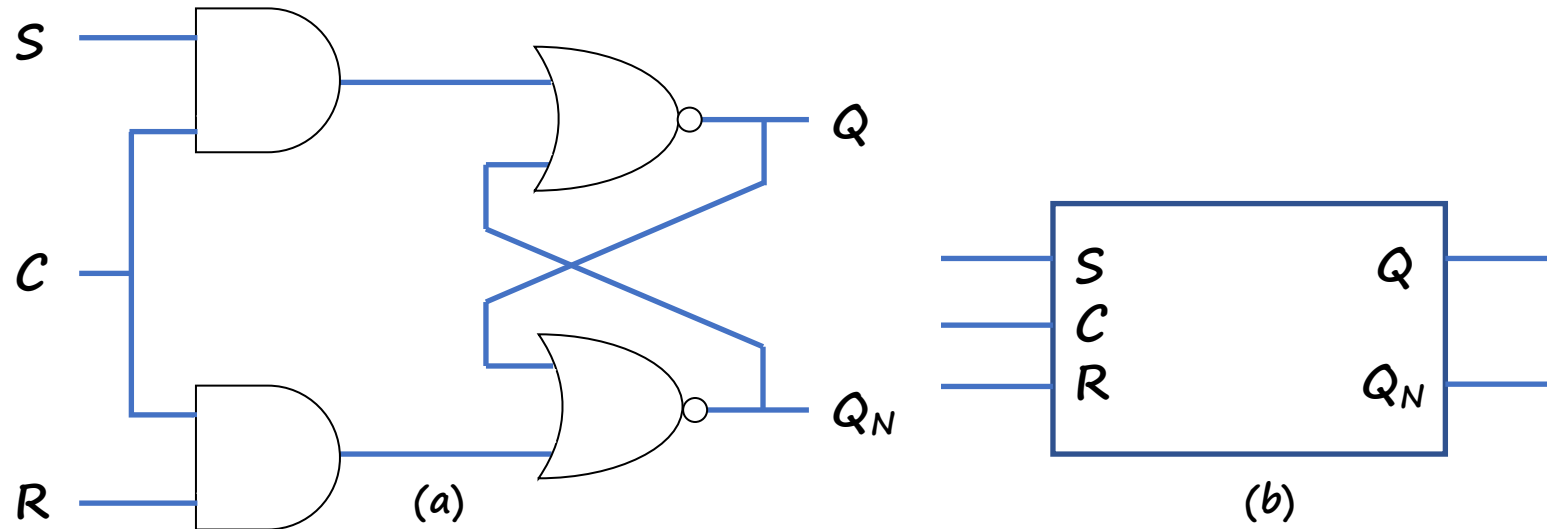
JK Tutucunun Zamanlama Özellikleri



Yukarıdaki zamanlama diyagramında, J ve K' nin yeterince uzun süre 1' de tutulması durumunda, tıpkı SR tutucuda olduğu gibi Q ve Q_N çıkışları sürekli olarak değerlerini tersine çevirmektedir. Bu durum istenmeyen bir durum olduğu için bu durumu ortadan kaldırmamanın bir yolunu bulmalıyız. Bu istenmeyen durum, JK flip-flopun kenar tetiklemesiyle veya master-slave türünden JK tutucu kullanarak ortadan kaldırılabilir.

İzin Girişli S-R Bilgi Saklama Elemanı (Tutucu)

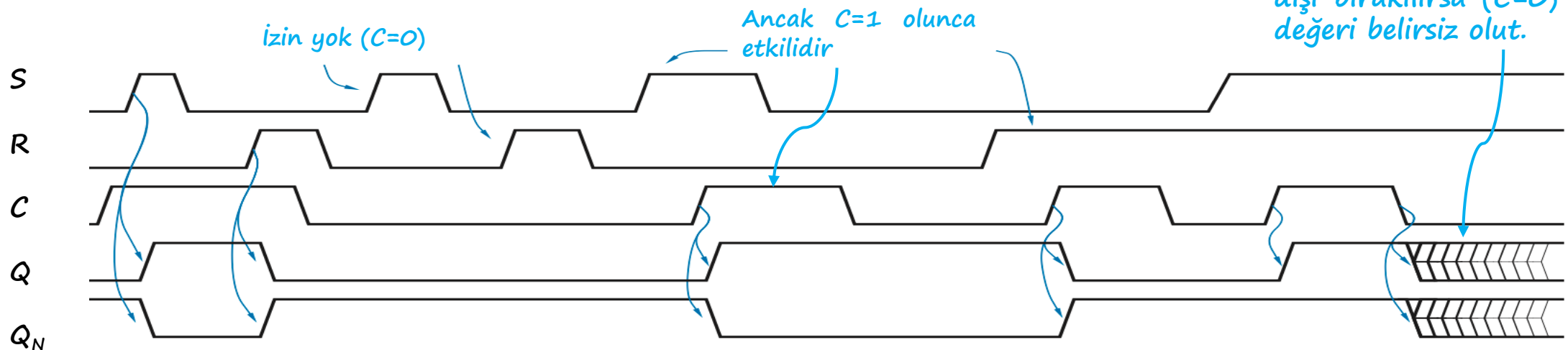
S ve R girişlerinin sadece istenen (izin verilen) zamanlarda etkili olabilmesi için bu girişlere VE (AND) kapıları bağlanır. Kontrol girişi (aşağıdaki şekilde C olarak etiketlenmiştir) bazen ENABLE, CLK veya G olarak adlandırılır



Tanım	C	S	R	$Q(n+1)$
TUTMA	0	X	X	$Q(n)$
TUTMA	1	0	0	$Q(n)$
SİLME	1	0	1	0
YAZMA	1	1	0	1
TANIMSIZ	1	1	1	Yasak

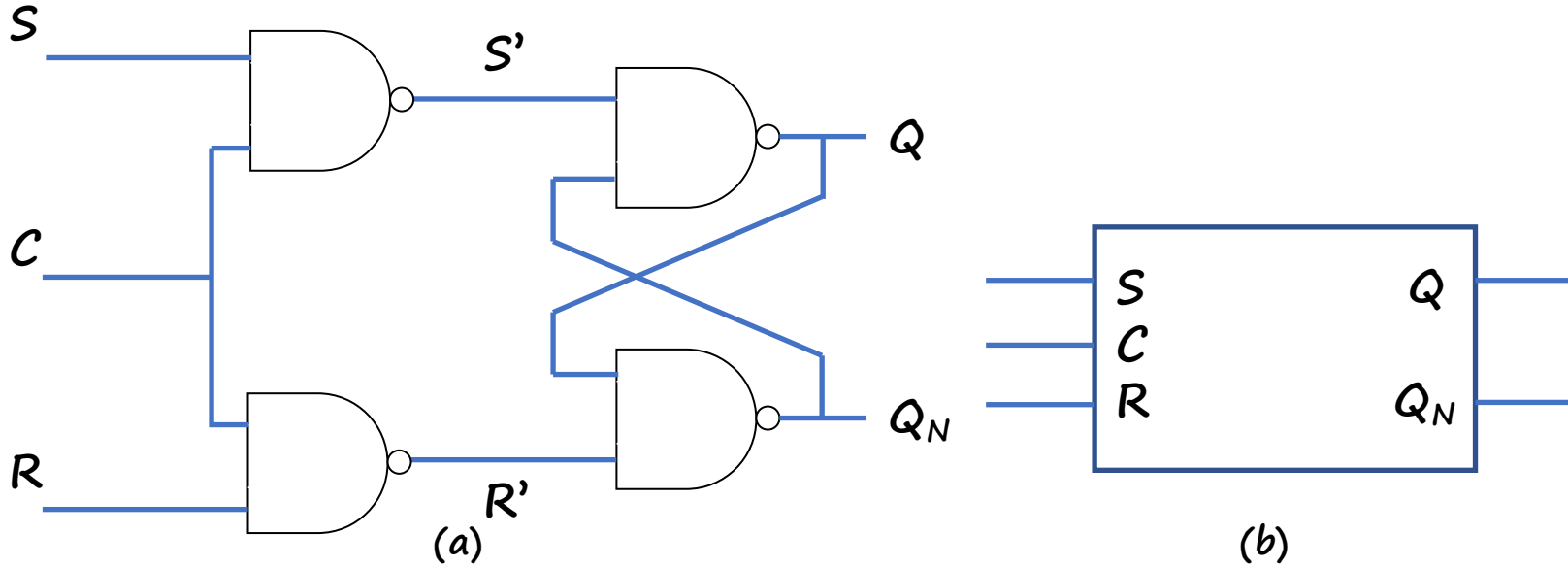
(c) Tutucaya yasak giriř (SR=11) uygulanırsa hem Q hem de QN ıkıřları 1 olur. Bu durumda izin giriři devre dıřı bırakılırsa (C=0) tutucu deęeri belirsiz olur.

İzin girişli SR tutucu: (a) NOR ve NAND kapılarını kullanan devre; (b) blok diyagramı; (c) doğruluk tablosu



İzin Girişli S-R Tutucu (Sadece NAND kapıları ile Tasarım)

İzin girişli S-R tutucu sadece NAND kapıları kullanılarak gerçekleştirilebilir.

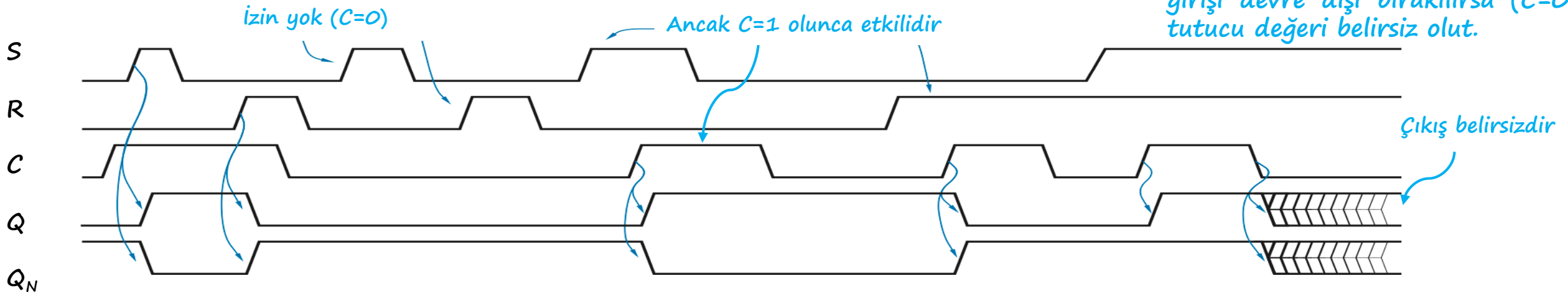


Tanım	C	S	R	Q(n+1)
TUTMA	0	X	X	Q(n)
TUTMA	1	0	0	Q(n)
SİLME	1	0	1	0
YAZMA	1	1	0	1
TANIMSIZ	1	1	1	Yasak

(c)

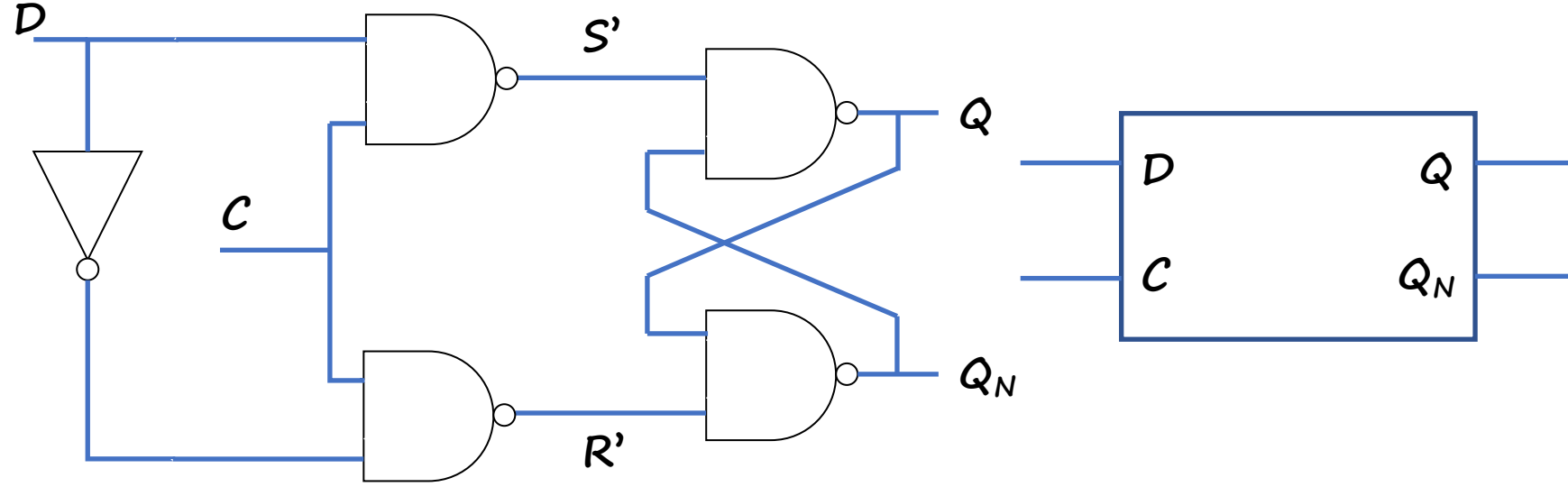
Tutucuya yasak giriş (SR=11) uygulanırsa hem Q hem de QN çıkışları 1 olur. Bu durumda izin girişi devre dışı bırakılırsa (C=0) tutucu değeri belirsiz olur.

İzin girişli SR tutucu: (a) NAND kapılarını kullanan devre; (b) blok diyagramı; (c) doğruluk tablosu



İzin Girişli D Tutucu

Kontrol girişi aktif yüksek (aşağıdaki şekilde olduğu gibi) veya aktif düşük olabilir.

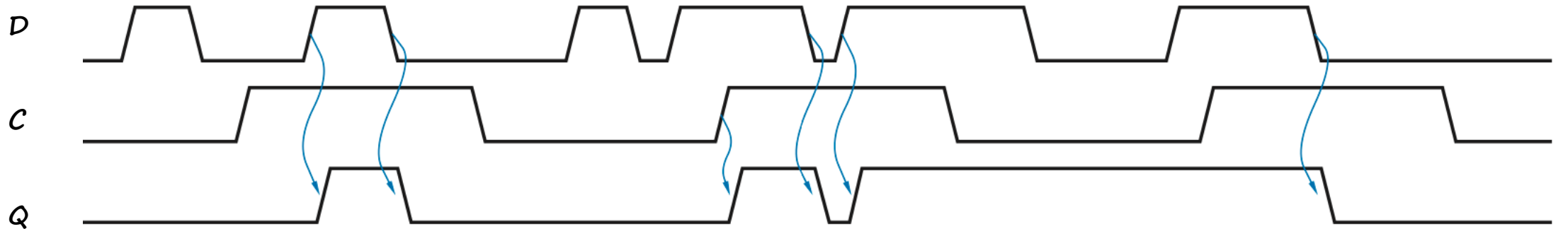


Tanım	C	D	Q(n+1)
TUTMA	0	X	Q(n)
SİLME	1	0	0
YAZMA	1	1	1

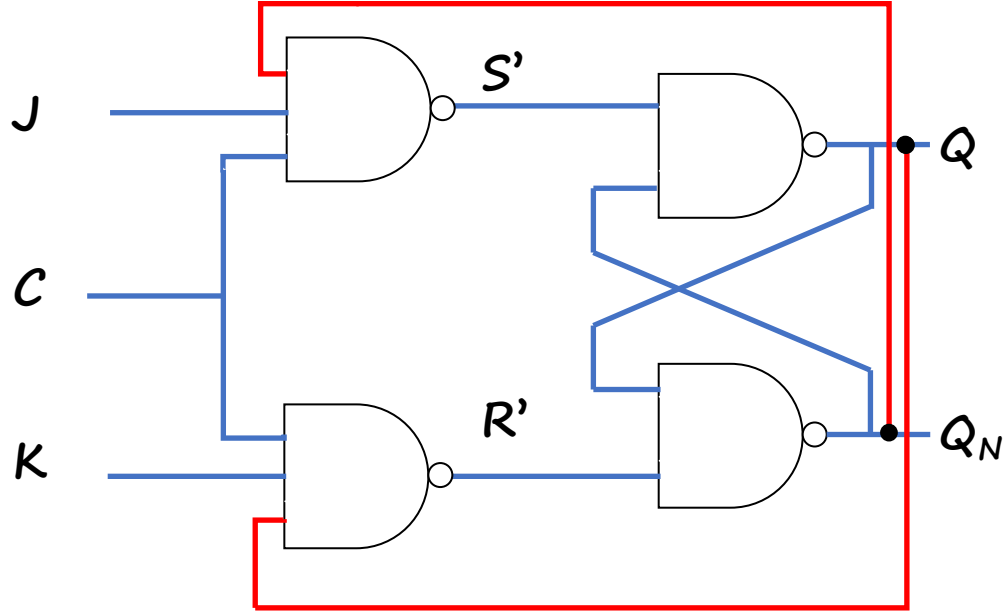
(c)

İzin girişli D tutucu: (a) NAND kapılarını kullanan devre; (b) blok diyagramı; (c) doğruluk tablosu

$Q=D$ çünkü $C=1$ ' dir (Transparan mod).



İzin Girişli JK Tutucu



(a)



(b)

Tanım	C	J	K	$Q(n+1)$
TUTMA	0	X	X	$Q(n)$
TUTMA	1	0	0	$Q(n)$
SİLME	1	0	1	0
YAZMA	1	1	0	1
TÜMLEME	1	1	1	$Q(n)'$

İzin girişli JK tutucu: (a) NAND kapılarını kullanan devre; (b) blok diyagramı; (c) doğruluk tablosu

Tutucularda Kullanılan Semboller

- C kontrol girişi üzerinde aktif bir seviye varsa, tutucu durum değiştirebilmektedir.
- Lojik-1 aktif seviye tutucular:
 - $C=\text{Lojik-1}$ ise tutucu durum değiştirebilir
 - Lojik-1 aktif seviye tutucular için kullanılan standart semboller:



- Lojik-0 aktif seviye tutucular:
 - $C=\text{Lojik-0}$ ise tutucu durum değiştirebilir
 - Lojik-0 aktif seviye tutucular için kullanılan standart semboller:



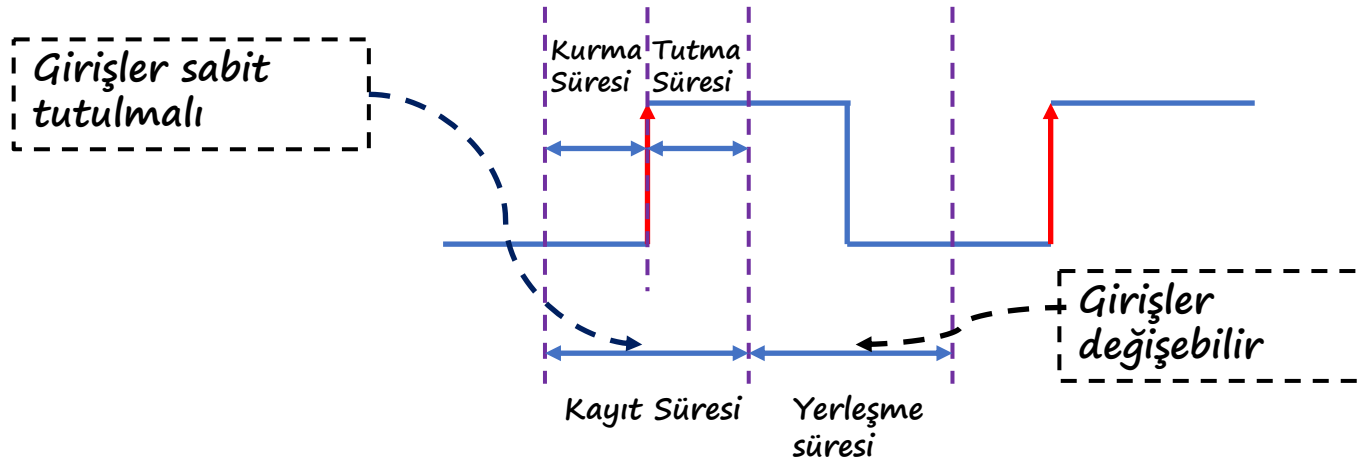
Tutucu (Latch) ve Flip-flop Farkı

- Bir Tutucunun bir saat sinyali ile tetiklenmesi söz konusu değildir. İzin girişi etkin olduğu sürece bu elemanın içeriği değiştirilebilir. Bu tip elemanlara Tutucu (latch) denir.
- Saat işareti ile tetiklenen saklama elemanlarına flip-flop denir.

Saat sinyalinin, devredeki kapıları senkronize eden periyodik bir kare dalga olduğunu unutmayın.

Kenar tetiklemeli elemanlar:

Bu elemanlar, aktif olarak saat sinyalinin bir kenarını (pozitif mantıkta yükselen kenar veya negatif mantıkta düşen kenar) kullanır..



Kurma süresi: "Set-up time"

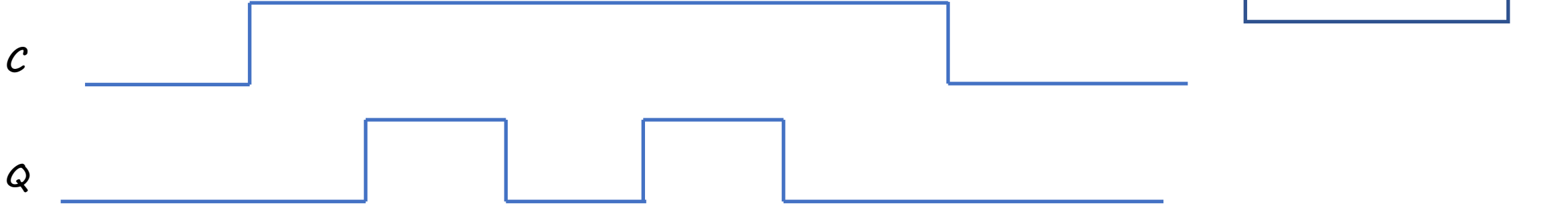
Tutma süresi: "Hold time"

Kayıt süresi, kurma ve tutma sürelerinin toplamından oluşur.

Ardışıl devrenin sağlıklı çalışması için bu süre boyunca girişlerin sabit kalması gerekir.

Tutucu Zamanlama Sorunları

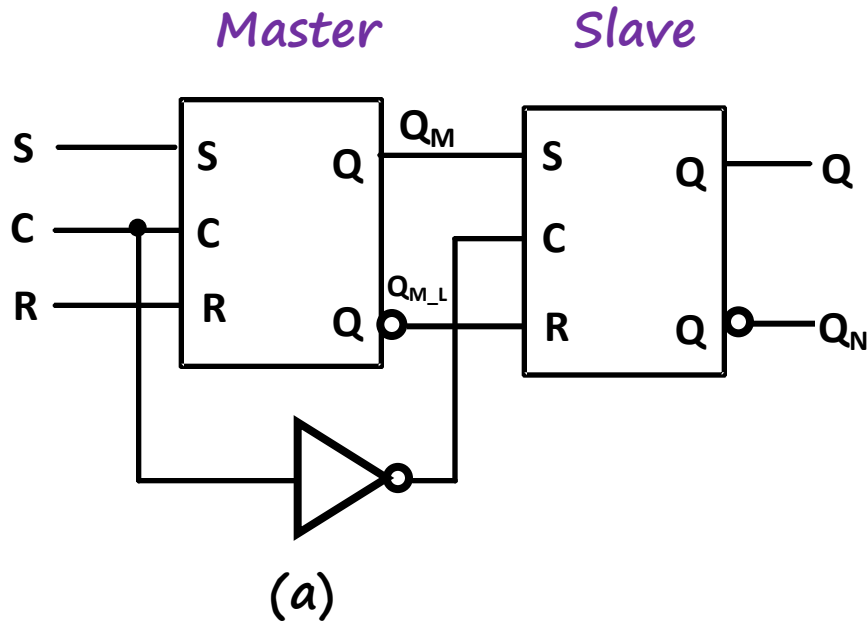
- Sağdaki devrede başlangıçta $Q = 0$ olduğunu varsayalım. Saat sinyali yüksek olduğunda, çıkış aşağıdaki dalga biçimine benzer (Çıkışta yüksek frekanslı bir işaret görülür)



- Anahtar Çıkarım:** "Tutucular aynı kontrol (saat) sinyaline sahip olduğunda, bir tutucunun çıkışı aynı veya başka bir tutucunun girişine doğrudan veya kombinasyonel devre yoluyla uygulanmamalıdır."
- Tercih edilen davranış:**
 - Saat darbesi boyunca çıkış yalnızca bir kez değişmelidir.
 - Çözüm: İstenmeyen geri beslemeyi ve çıkış değişikliklerini önlemek için devre çıkışı ile devre girişi arasındaki bağlantının koparılmasıdır.
- D-tutucu aşağıdakilerle değiştirilir :
 - Master (ana)-slave (uydu) flip-flop
 - Kenar tetiklemeli flip-flop

Ana (Master)-Uydu (Slave) tipi SR flip-flop

- Ana/uydu tipi flip-floplar darbe tetiklemeli (pulse-triggered) türden elemanlardır.
- $C = 1$ olduğunda giriş ilk flip-flopun çıkışında gözükür.
- $C = 0$ olduğunda giriş ikinci flip-flopun çıkışı değişir.
- Saat kaynağı giriş ve çıkış arasındaki bağlantıyı koparmıştır.



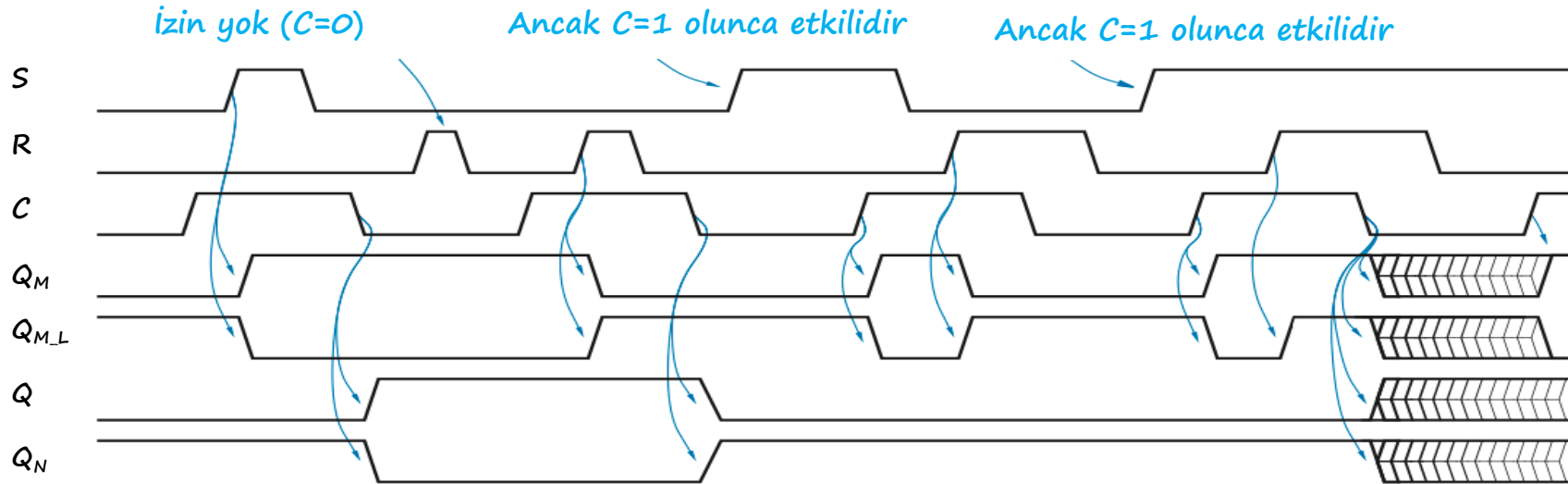
Tanım	S	R	C	Q	Q _N
TUTMA	X	X	0	Last Q	Last Q
TUTMA	0	0		Last Q	Last Q
SİLME	0	1		0	1
YAZMA	1	0		1	1
TANIMSIZ	1	1		-	-

(b)

Master-Slave SRFF (a) SR tutucuları kullanan devre; (b) doğruluk tablosu

Ana - Uydu tipi SR flip-flop (devam)

- Bu tip flip-flopun içeriği (çıkışı) sadece saat işaretinin inen kenarında değişir.
- Ancak flip-flopun alacağı değer saat işaretinin 1 olduğu süre boyunca belirlenir.
- Aşağıda gösterildiği gibi $C=1$ olduğunda, S üzerindeki bir darbe Master tutucunun çıkışını 1 yapar ve R üzerindeki bir darbe ise Master tutucuyu sıfırlar.
- Master tutucunun çıkışına bağlı olarak Slave tutucunun çıkışındaki veri ise düşen kenarda değişmektedir
- Devre daha çok, $C=1$ olduğunda tüm aralık için girişlerini takip eden, ancak bunları yalnızca $C=1$ olduğunda saklayan bir tutucu gibi davranır.



Ana - Uydu tipi SR flip-flop (devam)

- Bu türden flip-flop gerçekten kenar tetiklemeli bir eleman olmadığı için aşağıda gösterilen sembol kullanılıyor.

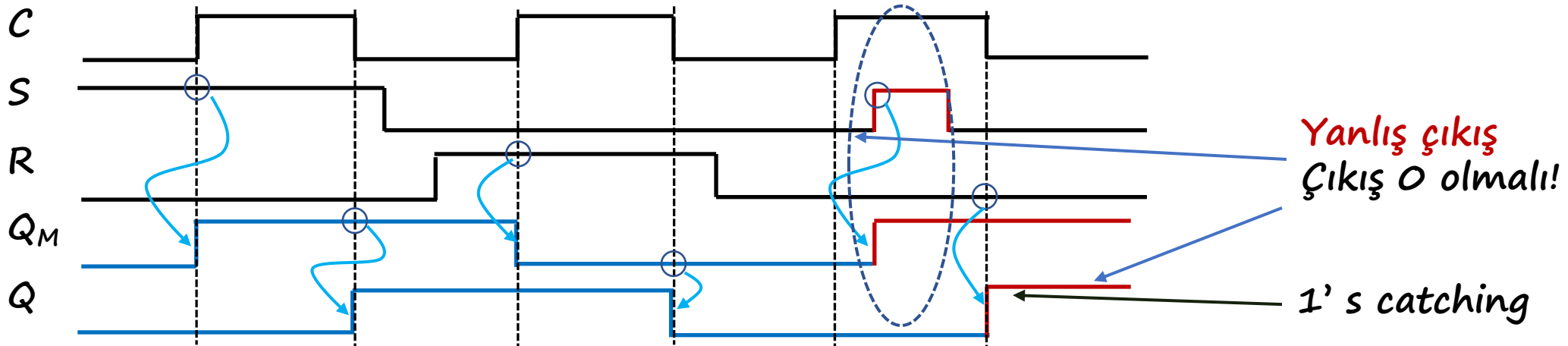


Lojik sembol

- Bunun yerine, C girişinin durum değiştirene kadar çıkış değerinin değişmediğini belirtmek için ertelenmiş bir çıkış göstergesi kullanılır.
- Devre bazen darbe tetiklemeli flip-flop olarak adlandırılır.
- Saat kaynağının düşen kenarından hemen önce $S=R=1$ yapılırsa, her iki çıkışın 1 olmasını sağlar ($Q_M = Q_{M_L} = 1$)

Ana - Uydu tipi SR flip-flop Problemi

- Flip-flop çıkışındaki değişiklik, saat kaynağı tarafından geciktirilir, bu da devreyi yavaşlatır
- Doğru çalışma için saat darbesi süresince Master çıkışı değişmemelidir.
 - $CLK=1$, $Q_M=0$ ve $R=0$ iken $S = 0 \rightarrow 1 \rightarrow 0$ değişimini yapsın
 - Master tutucu çıkışı $0 \rightarrow 1$ değişimini yapar
 - Düşen kenar meydana geldiğinde ise Slave tutucu çıkışı 1 olur (1's catching-1 yakalama)
 - $CLK=1$, $Q_M=1$ ve $S=0$ iken $R = 0 \rightarrow 1 \rightarrow 0$ değişimini yapsın
 - Master tutucu çıkışı $1 \rightarrow 0$ değişimini yapar
 - Düşen kenar meydana geldiğinde ise Slave tutucu çıkışı 0 olur (0's catching-0 yakalama)

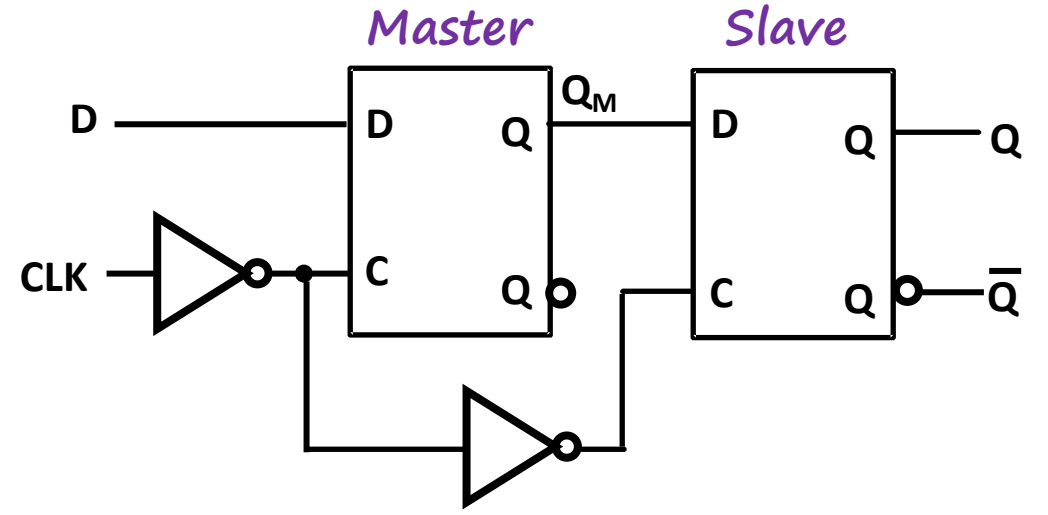


Çözüm: Flip-Flop

- Master-slave FF yerine kenar tetiklemeli FF kullanın
- Kenar tetiklemeli bir flip-flop, yalnızca saat sinyalinin 0→1 ya da 1→0 geçişi sırasında durum değiştirir.
- Kenar tetiklemeli bir flip-flop doğrudan elektronik devre seviyesinde oluşturulabilir
- Kenar tetiklemeli davranış sergileyen bir master-slave DFF kullanılabilir.

Master-Slave D Flip-Flop

- Pozitif kenar tetiklemeli bir DFF sağda gösterilmiştir.
- Yalnızca D girişini örnekler ve CLK sinyalinin yükselen kenarında çıkışını değiştirir.
- İlki 'master-ana', ikincisi de 'slave-uydu' olarak adlandırılan iki D tutucudan oluşur.
- $CLK=0$ olduğunda ana tutucu etkindir ve girişi takip eder.
- $CLK=1$ olduğunda ise ana tutucu etki değildir, çıkışını uydu tutucuya aktarır.
- Uydu tutucu $CLK=1$ olduğunda her zaman etkindir, ve çıkışını yalnızca CLK sinyalinin yükselen kenarında değiştirir.
- D flip-flopun sembolü üzerindeki CLK girişindeki üçgen, dinamik giriş göstergesi olarak adlandırılır ve kenar tetiklemeli davranışı gösterir.



Pozitif kenar tetiklemeli DFF

Doğruluk tablosu:

D	CLK	$Q(n+1)$	$Q_N(n+1)$	Tanım
0		0	1	SİLME
1		1	0	YAZMA
x	0	$Q(n)$	$Q_N(n)$	TUTMA
x	1	$Q(n)$	$Q_N(n)$	TUTMA


Yükselen kenar

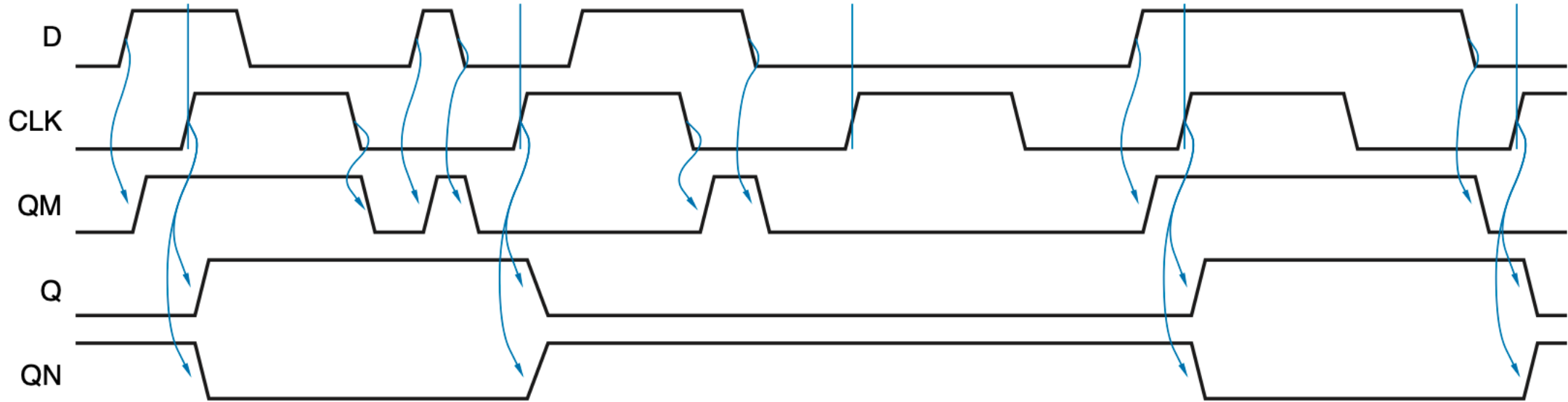


Lojik sembol

Pozitif (yükselen) kenar tetiklemeli D tipi Flip-flop

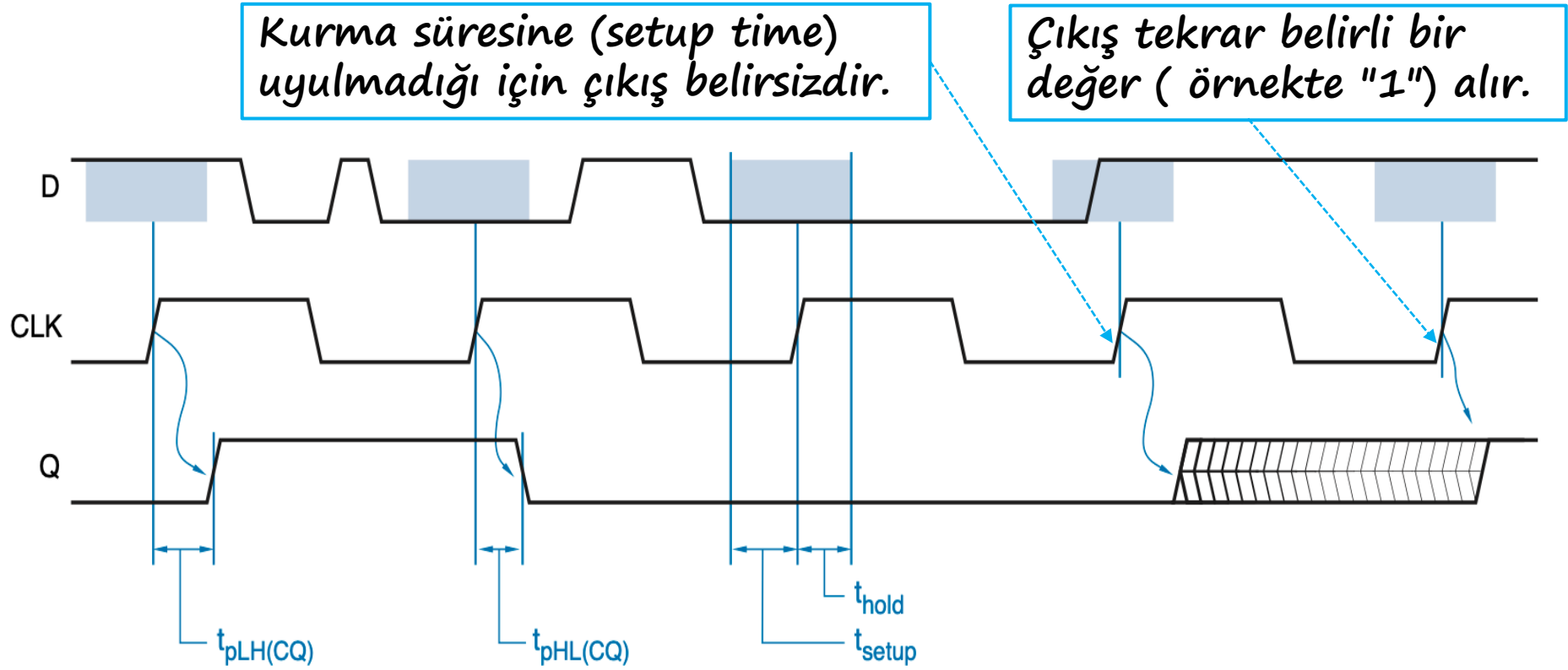
Aşağıdaki şekil pozitif kenar tetiklemeli D flip-flopun fonksiyonel davranışını göstermektedir.

- QM sinyali ana tutcunun çıkışıdır
- QM yalnızca CLK=0 olduğunda değişir
- CLK 0→1 geçişini yaptığında, QM' nin mevcut değeri Q' ya aktarılır ve CLK=1 olduğu sürece QM tekrar değişmez.



Pozitif kenar tetiklemeli bir D flip-flopun fonksiyonel davranışı

Pozitif kenar tetiklemeli D tipi flip-flopunun zamanlama özellikleri



$t_{pLH(CQ)}$: Etkin kenardan sonra çıkışın 0-1 geçişi yapması için geçen süre.

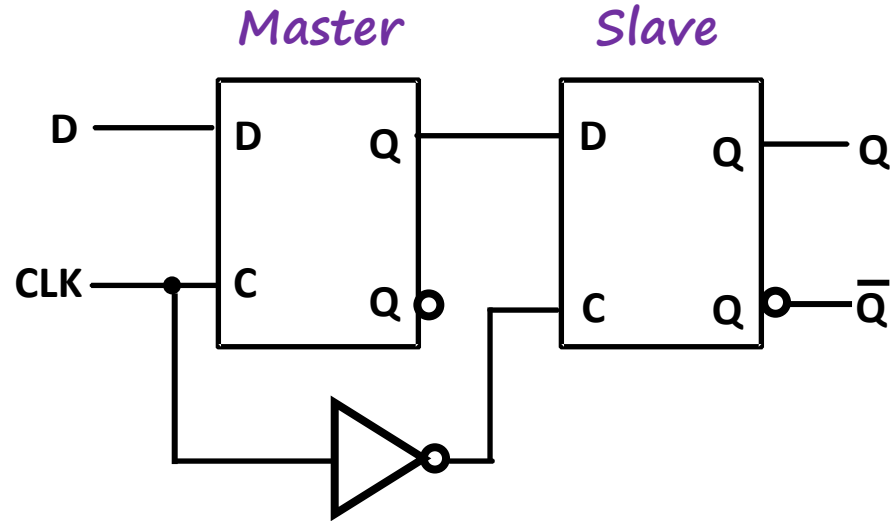
$t_{pHL(CQ)}$: Etkin kenardan sonra çıkışın 1-0 geçişi yapması için geçen süre.

t_{setup} : Etkin kenardan önce girişin sabit kalması gereken süre.

t_{hold} : Etkin kenardan sonra girişin sabit kalması gereken süre.

Negatif (inen) kenar tetiklemeli D tipi Flip-flop

Saat işaretinin inen kenarlarında D girişindeki veri flip-flopa yazılır



(a)

 Düşen kenar

D	CLK	$Q(n+1)$	$Q_N(n+1)$	Description
0		0	1	SİLME
1		1	0	YAZMA
x	0	$Q(n)$	$Q_N(n)$	TUTMA
x	1	$Q(n)$	$Q_N(n)$	TUTMA

(b)

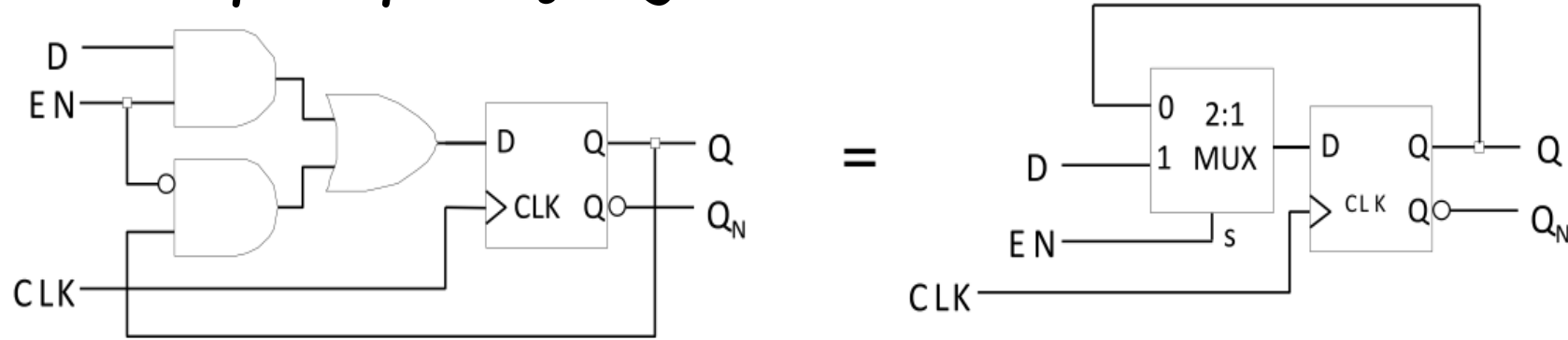


(c)

Negatif kenar tetiklemeli D tipi Flip-flop: (a) D Tutucu kullanan devre; (b) doğruluk tablosu; (c) lojik sembol

Kenar tetiklemeli ve izin giriřli D tipi Flip-flop

Flip-floplarda da izin giriři (EN) (enable) bulunabilir. Flip-flopun ieriğinin deėiřtirilebilmesi iin izin giriři (EN) etkin olmalıdır. Aksi durumda flip-flopun ieriėi korunur.



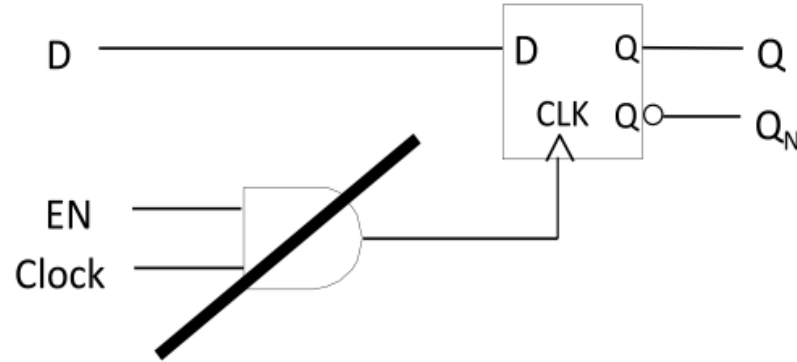
Doėruluk tablosu:

D	EN	CLK	$Q(n+1)$	$Q_N(n+1)$
0	1		0	1
1	1		1	0
x	0		$Q(n)$	$Q_N(n)$
x	x	0	$Q(n)$	$Q_N(n)$
x	x	1	$Q(n)$	$Q_N(n)$



Kenar tetiklemeli ve izin girişli D tipi Flip-flopu (devam)

Flip flopun saat girişine (CLK) bir VE kapısı bağlayarak daha basit bir devre elde edebilir miyiz?



Bu uygun bir çözüm değildir.

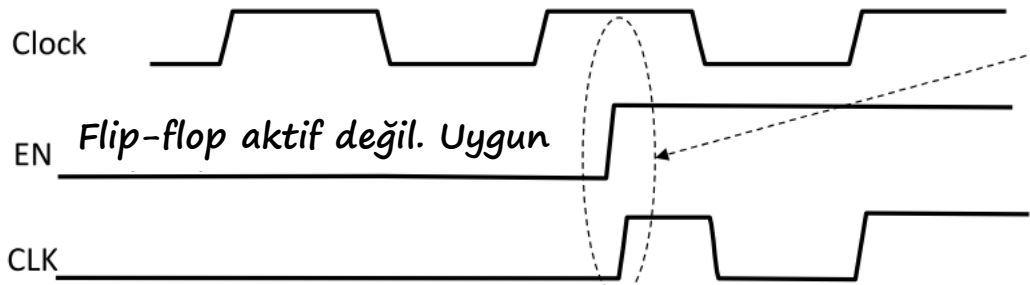
Bu devrede aşağıdaki sorunlar vardır.

1. Flip flopun saat girişindeki gecikmeyi arttırmak iyi bir değildir. İbu durum senkronizasyon sorunlarına neden olabilir.

2. Eğer Saat=1 ve EN=0 ise CLK=0 olur ve flip flop çalışmaz (uygun bir durum).

Ancak Saat=1 iken EN=1 olursa flip-flopun CLK girişinde bir aktif kenar oluşur ve flip flop D girişindeki değeri işler.

Halbuki saat işareti o anda 1'dir; 0-1 geçişi yapılmamıştır.



Sorun:

Saat işaretinde gerçekte çıkan kenar yok ancak flip-flopun saat girişinde çıkan kenar oluşuyor.

T Flip-Flop

Lojik sembol



Karakteristik tablosu:

CLK	T	Q(n)	Q(n+1)	Tanım
	0	0	0	TUTMA
	0	1	1	TUTMA
	1	0	1	TÜMLEYEN
	1	1	0	TÜMLEYEN
	X	X	Q(n)	TUTMA

Doğruluk tablosu:

T	Q(n+1)	Tanım
0	Q(n)	TUTMA
1	Q(n)'	TÜMLEYEN

0, 1 ya da düşen kenar

Yükselen kenar

Uyarma tablosu :

Q(n)	Q(n+1)	T	Tanım
0	0	0	TUTMA
1	1	0	TUTMA
0	1	1	TÜMLEYEN
1	0	1	TÜMLEYEN

Q(n) \ T	$\bar{T}$	T
$\bar{Q}(n)$	0 0	2 1
Q(n)	1 1	3 0

Karakteristik denklem:

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

Flip Flop Asenkron Girişleri

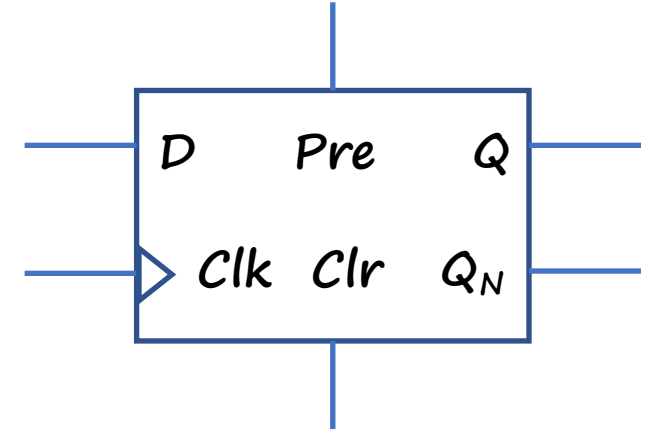
Flip-floplar genellikle flip-flopu saatten bağımsız (asenكرون) bir başlangıç durumuna ayarlamak için kullanılabilen ek girişlere sahiptir.

- Pre (Preset), flip-flopa "1" yazmak için kullanılır. 'Preset (Pre)' girişine uygulanan lojik "1", flip-flopun Q çıkışını "1" olarak ayarlayacaktır.
- CLR (Clear) flip-flopa "0" yazmak için kullanılır. Clear (CLR) girişine uygulanan lojik "1", flip-flopun Q çıkışını sıfırlayacaktır (0).

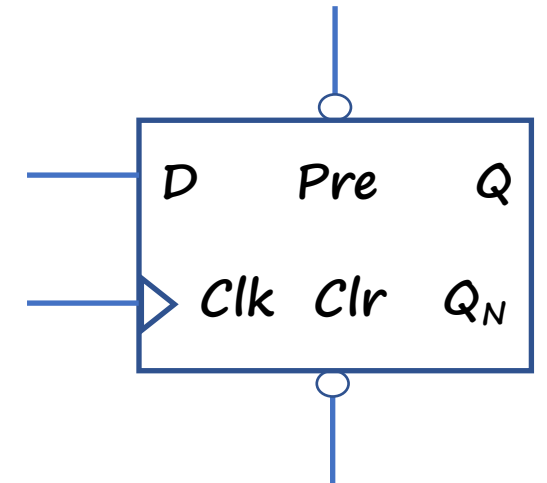
Bu asenkron girişler saat ve D girişlerini geçersiz kılar. Yani, silme girişine (CLR) uygulanan lojik "1", D ve saat değerlerinden bağımsız olarak flip-flopu sıfırlayacaktır.

Bu asenkron girişler, flip flopa güç verdikten sonra flip flop' a başlangıç değerleri atamak için kullanılabılır.

'Pre' ve/veya 'CLR' girişleri için negatif mantık kullanan cihazlar da vardır. Örneğin böyle bir cihazda flip-flop' un Q çıkışını "0" yapmak için 'CLR' girişine "0" uygulanmalıdır.



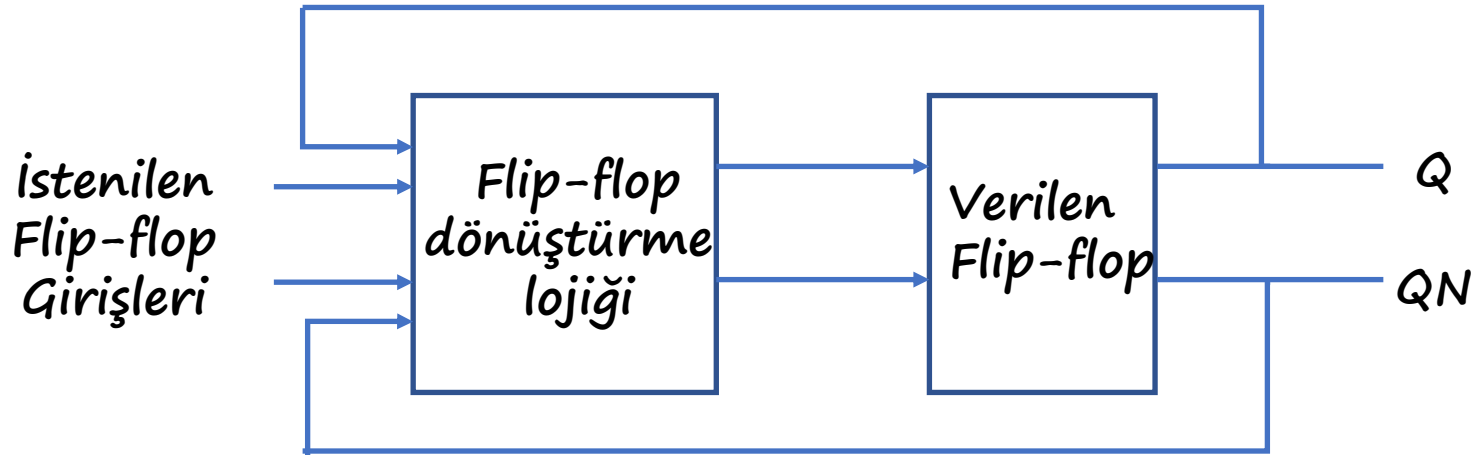
Pozitif asenkron girişli DFF



Negatif asenkron girişli DFF

FLIP FLOPLAR ARASINDA DÖNÜŞÜM

Aşağıda görülen şekilde, yeni giriş tanımlarını, verilen flip-flopun istenen flip-flop gibi çalışmasına neden olacak giriş kodlarına dönüştürmek için dönüştürme mantığının tasarlanması gerektiği görülmektedir. Flip-flop dönüştürme mantığını tasarlamak için, her iki flip-flop için uyarma tablosunu birleştirmemiz gerekmektedir. Ardından tabloda istenilen FF girişleri ve Q giriş olarak ve mevcut (verilen) flip-flop girişleri ise çıkış olarak gösterilmelidir.



Bir flip-flopun başka bir türe dönüştürülmesi için kullanılan genel bir model

Bir S-R Flip-flopun D Flip-flopa dönüştürülmesi:

Adım 1: Her iki FF için bir uyarma tablosu hazırlanır.

Çıkış		SRFF (Mevcut) Girişleri		DFF (İstenilen) Girişleri
Q(n)	Q(n+1)	S	R	D
0	0	0	d	0
0	1	1	0	1
1	0	0	1	0
1	1	d	0	1

Adım 2: Dönüştürme tablosunda, istenilen FF girişleri ve Q giriş olarak ve mevcut (verilen) FF girişleri ise çıkış olarak gösterilmelidir

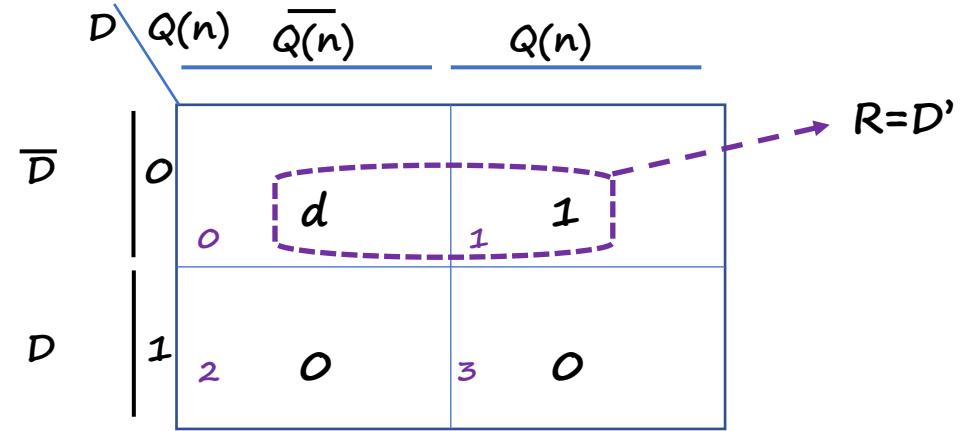
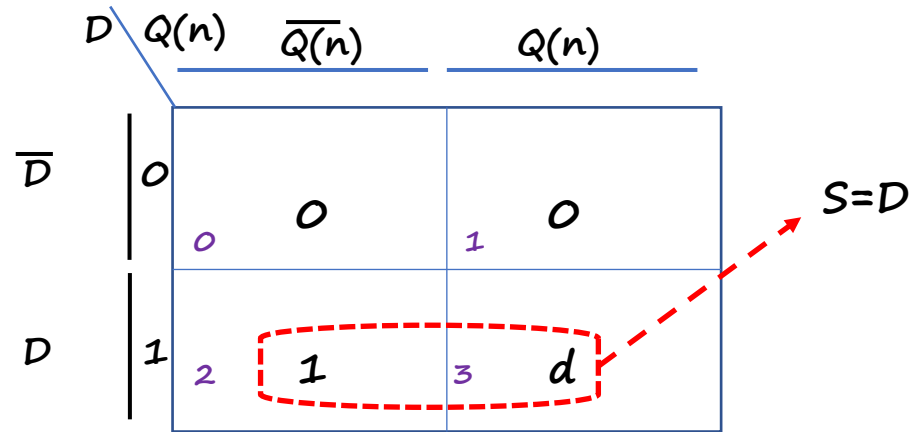
DFF (İstenilen) girişleri	Çıkış		SRFF (Mevcut) Girişleri	
D	Q(n)	Q(n+1)	S	R
0	0	0	0	d
1	0	1	1	0
0	1	0	0	1
1	1	1	d	0

Dönüştürme tablosu

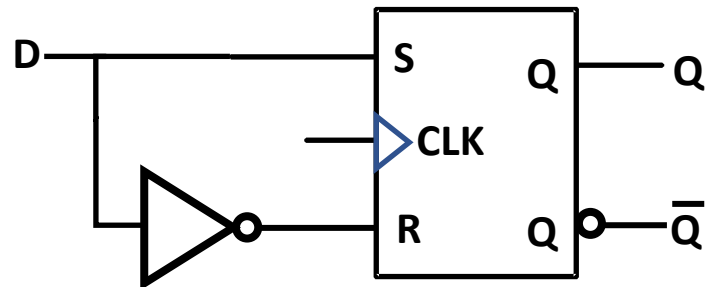
Bir S-R Flip-flopun D Flip-flopa dönüştürülmesi (devam):

Adım 3: Verilen flip-flopun girişleri için ifadeleri basitleştirin.

Bu durumda, K-haritası gibi uygun basitleştirme teknikleri kullanılarak S ve R girişleri için D ve Q_n cinsinden mantıksal ifadelere ulaşılması gerekir.



Karnaugh haritalarının yardımıyla sadeleştirme sonucunda $S = D$ ve $R = D'$ elde ederiz.



Bir JK Flip-flopun T Flip-flopa dönüştürülmesi:

TFF Girişleri	Çıkış		JKFF Girişleri	
T	$Q(n)$	$Q(n+1)$	J	K
0	0	0	0	d
1	0	1	1	d
1	1	0	d	1
0	1	1	d	0

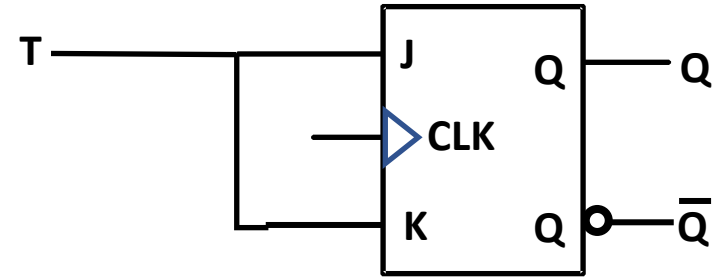
Dönüştürme tablosu

T	$Q(n)$	$\overline{Q(n)}$	$Q(n)$
$\overline{T}$	0	d	0
T	1	d	1

K=T

T	$Q(n)$	$\overline{Q(n)}$	$Q(n)$
$\overline{T}$	0	0	d
T	1	1	d

J=T



JKFF' den TFF elde edilmesi