

# Sıralı Mantık (Sequential Logic) Devreleri

Latch, Flip-Flop, Devre Analizi

# Latch ve Flip-Flop

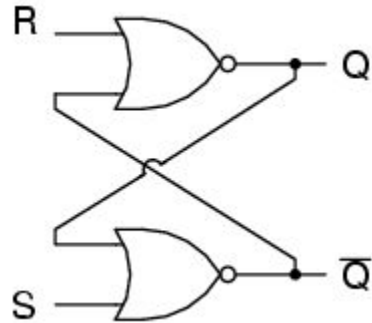
# Sıralı Mantık Devreleri

- Kombinasyonel devrelerin çıktısını sadece girdileri belirler. Aynı girdi için her zaman aynı çıktıyı verirler.
- Sıralı (sequential) mantık devrelerinin çıktısını girdileri ve/veya bir önceki durumları (state) belirler. Aynı girdi için farklı zamanlarda farklı değerler verebilirler.
- Bu bakımdan sıralı mantık devrelerinin “state”i olduğunu söyleyebiliriz.
- Sayı sayan bir devre sıralı mantık devrelerine örnek verilebilir.
- Devrelere bellek / state kazandırmak için halkasal bağlantılar kullanılabilir.

# SR Latch

Belleği olan temel bileşenlerden biridir. Verilen değeri saklayabilir.

- **S**: Set. Q değerini 1 yapmak için kullanılır.
- **R**: Reset. Q değerini 0 yapmak için kullanılır.



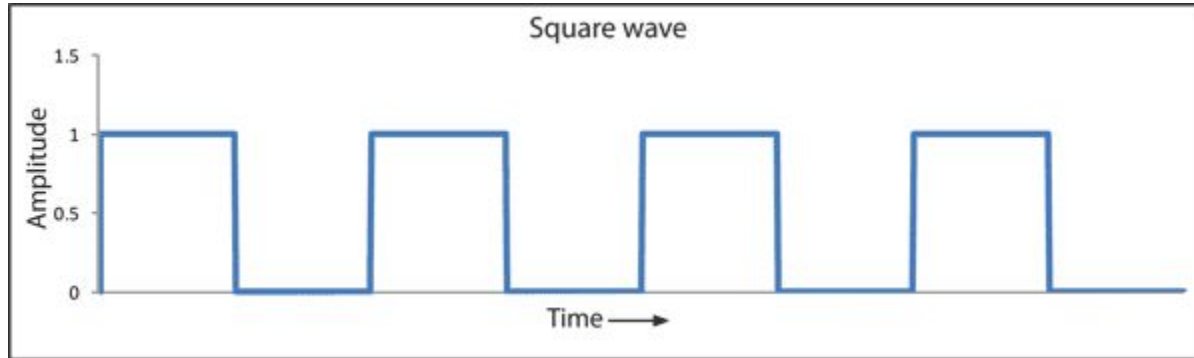
S	R	Q	$\bar{Q}$
0	0	latch	latch
0	1	0	1
1	0	1	0
1	1	0	0

# Sync ve Async Devreler

- Asynchronous (eşzamansız) devreleri kontrol etmek için bir sinyal kullanılmaz. Devre bir süre sonra denge durumuna ulaştığında sonucu üretmiş olur. Hızlıdır, tasarlaması zordur. Devre hiçbir zaman dengeye ulaşmayabilir. Çipin üretim kalitesine göre bazı devre elemanları daha hızlı sonuç üretebilir, bu durum aynı devrenin farklı çiplerde üretim kalitesi nedeniyle farklı sonuçlar üretmesine sebep verebilir (race condition).
- Async devrelere bir kontrol sinyali uygulanarak devrenin farklı elemanlarının eş zamanlı çalışması sağlanabilir. Böylece race condition engellenmiş olur. Bu devrelere Synchronous (eş zamanlı) devre, sinyale Clock denir. Async devrelere göre daha yavaş çalışır ancak tasarlamak daha kolaydır.

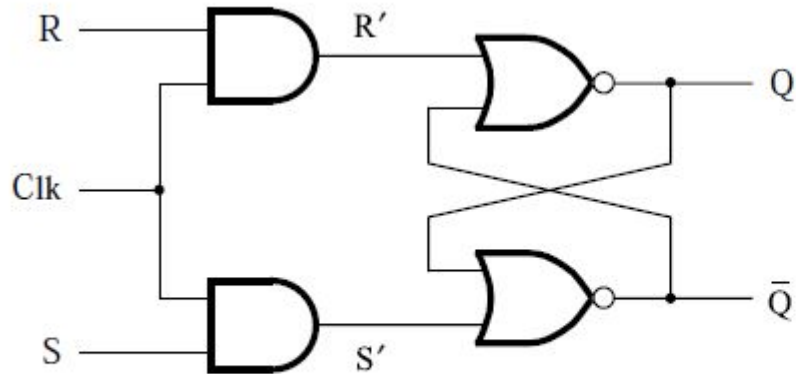
# Clock

- Genel olarak bir kare dalgadır. “Basitçe” clock high iken devre çalışıyor, low iken devre çalışmıyor gibi düşünülebilir. Ancak bazı devreler clock sinyalinin **HIGH → LOW** veya **LOW → HIGH** geçişinde çalışır.



# SR Flip-Flop

SR Latch Clock sinyali kullanacak şekilde tasarlandığında SR Flip-Flop elde edilir. Clk girişinden clock sinyali uygulanabilir.  **$S = 1, R = 1$  sinyalleri kullanılmaz.**

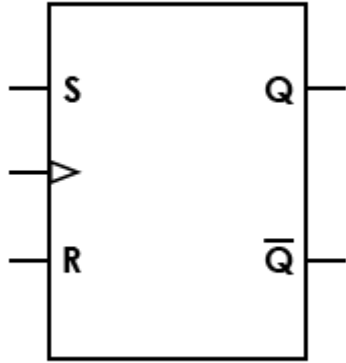


(a) Circuit

Clk	S	R	$Q(t+1)$
0	x	x	$Q(t)$ (no change)
1	0	0	$Q(t)$ (no change)
1	0	1	0
1	1	0	1
1	1	1	x

(b) Characteristic table

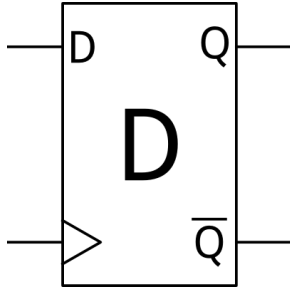
# SR Flip-Flop



- Solda devre sembolü gösterilmiştir.
- S ve R girişlerinin ortasındaki giriş Clock girişidir. Clock girişinin olduğu yerdeki > simgesi bu flip-flop'un **edge-triggered** olduğunu gösterir.
- **Level Triggered:** Flip-flop değeri clock sinyali HIGH iken değişir.
- **Edge Triggered:** Flip-flop değeri clock sinyali LOW'dan HIGH'a geçerken değişir.



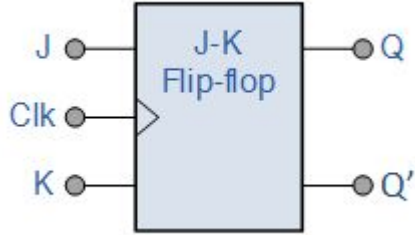
# D Flip-Flop



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

- Çok basit bir flip-flop'tur. Clock sinyalinin geçişinde D girdisindeki değeri alır ve Q çıktısı olarak sunar.
- Doğruluk tablosu (Truth Table) sol altta verilmiştir. Q çıktının şimdiki değerini,  $Q(t+1)$  çıktının bir sonraki Clock sinyalinden sonraki değerini gösterir. Q yerine  $Q_n$ ,  $Q(t+1)$  yerine  $Q_{n+1}$  de kullanılabilir.

# JK Flip-Flop



- SR flip-flop'ta S ve R değerlerinin 3 kombinasyonu kullanılır:
  - $S = 0, R = 0$ : Değeri değiştirme
  - $S = 1, R = 1$ : Değeri 1 yap
  - $S = 0, R = 1$ : Değeri 0 yap
- JK flip-flop her iki girdinin 1 olduğu durumun da kullanılmasını sağlar.
  - $J = 1, K = 1$  : Değeri tersine çevir

# JK Flip-Flop Truth Table

Truth Table sadece girdilere göre çıktıları gösterir.

Input		Output
$J_n$	$K_n$	$Q_{n+1}$
0	0	$Q_n$
0	1	0
1	0	1
1	1	$Q'_n$

Q'yu değiştirme

Q'yu 1 yap

Q'yu 0 yap

Q'yu tersine çevir

# JK Flip-Flop Characteristic Table

Characteristic Table girdilere ve güncel state'e göre çıktıları gösterir.

Input		Present Output	Next Output
J	K	$Q_n$	$Q_{n+1}$
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	1
1	1	1	0

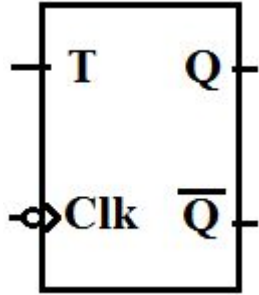
# JK Flip-Flop Excitation Table

Excitation Table verilen bir state geçişi için kullanılması gereken girdileri gösterir.

Q	Q <sup>+</sup>	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

**Excitation of J-K flip-flop**

# T Flip-Flop



- Basit bir flip-flop'tur. T girdisi 1 ise Q'nun değeri tersine döner. Eğer T 0 ise Q'nun değeri değişmez.

$T$	$Q_n$	$Q_{n+1}$
0	0	0
0	1	1
1	0	1
1	1	0

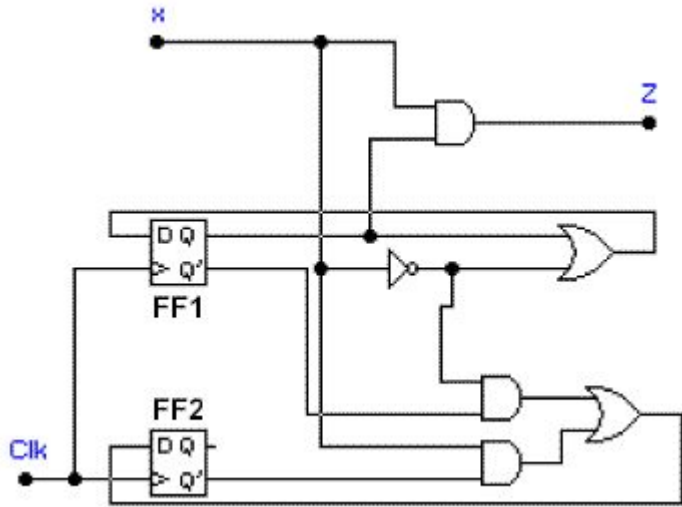
# Devre Analizi

# State Table ve State Diagram

- **State Table:** Güncel state'i verilen bir sıralı mantık devresinin tüm girdi kombinasyonları için sıradaki state'ini ve çıktısını gösterir.
- **State Diagram:** State geçişlerinin görselleştirilmesini ve daha kolay anlaşılmasını sağlar.



## Örnek



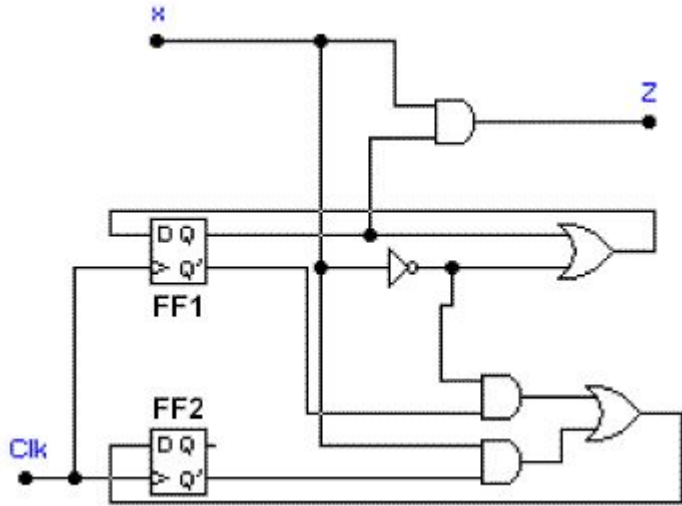
Soldaki devrenin State Table ve Diagram'larını hazırlayalım.

Girdiler : x

Stateler :  $Q_1$  ve  $Q_2$

Çıktılar:  $z$

## Örnek



Bunun için çıktılarının ve Flip-Flop girdilerinin Boolean denklemlerini bulmamız gerekir. Denklemler aşağıda bulunmuştur.

$$\mathbf{Z} = \mathbf{X} * \mathbf{Q}_1$$

$$\mathbf{D}_1 = \mathbf{x}' + \mathbf{Q}_1$$

$$\mathbf{D}_2 = \mathbf{x} * \mathbf{Q}_2' + \mathbf{x}' * \mathbf{Q}_1'$$

# Örnek

Denklemler kullanılarak State Table hazırlanabilir:

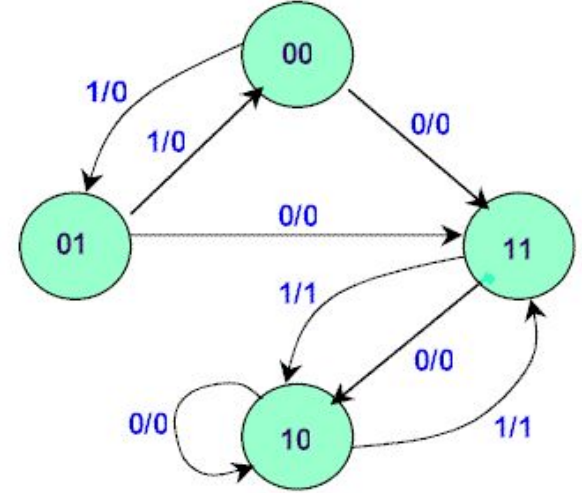
1. Güncel state'leri listele
2. Tüm girdi kombinasyonları için sıradaki state ve çıktıları bul

Present State $Q_1Q_2$	Next State		Output	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
0 0	1 1	0 1	0	0
0 1	1 1	0 0	0	0
1 0	1 0	1 1	0	1
1 1	1 0	1 0	0	1

# Örnek

State Table kullanılarak State Diagram çizilir:

1. Her state için bir yuvarlak çiz ve içine state'i yaz
2. Her input kombinasyonu için güncel state'ten sonraki state'e bir ok çiz (state değişmiyorsa state'in kendisine dönen bir ok çiz)
3. Okun üstüne girdi (x) ve ona karşılık gelen çıktı (y) değerlerini x/y formatında yaz.



# Flip-Floplar'ın State Diagram'ları

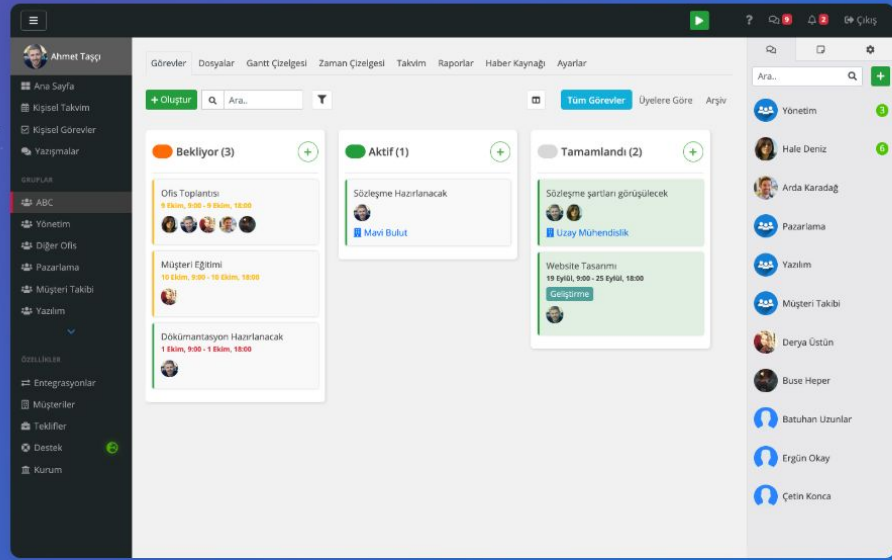
NAME	STATE DIAGRAM
SR	<pre> graph LR     Q0((Q = 0))     Q1((Q = 1))     Q0 -- "S,R=0,0" --&gt; Q0     Q0 -- "S,R=1,0" --&gt; Q1     Q1 -- "S,R=0,1" --&gt; Q0     Q1 -- "S,R=0,0" --&gt; Q1         </pre>
JK	<pre> graph LR     Q0((Q = 0))     Q1((Q = 1))     Q0 -- "J,K=0,0" --&gt; Q0     Q0 -- "J,K=1,0 or 1,1" --&gt; Q1     Q1 -- "J,K=0,1 or 1,1" --&gt; Q0     Q1 -- "J,K=0,0" --&gt; Q1         </pre>
D	<pre> graph LR     Q0((Q = 0))     Q1((Q = 1))     Q0 -- "D=1" --&gt; Q0     Q0 -- "D=0" --&gt; Q1     Q1 -- "D=1" --&gt; Q0     Q1 -- "D=0" --&gt; Q1         </pre>
T	<pre> graph LR     Q0((Q = 0))     Q1((Q = 1))     Q0 -- "T=0" --&gt; Q0     Q0 -- "T=1" --&gt; Q1     Q1 -- "T=1" --&gt; Q0     Q1 -- "T=0" --&gt; Q1         </pre>

\* D flip-flopta soldaki halkanın değeri  $D = 0$  olacak

Reklamlar

# İşinizi Kolayca Yönetin

Cubicl görev yönetimi, proje planlama, takım işbirliği  
ve müşterilerinizi tek bir platformda birleştirir.



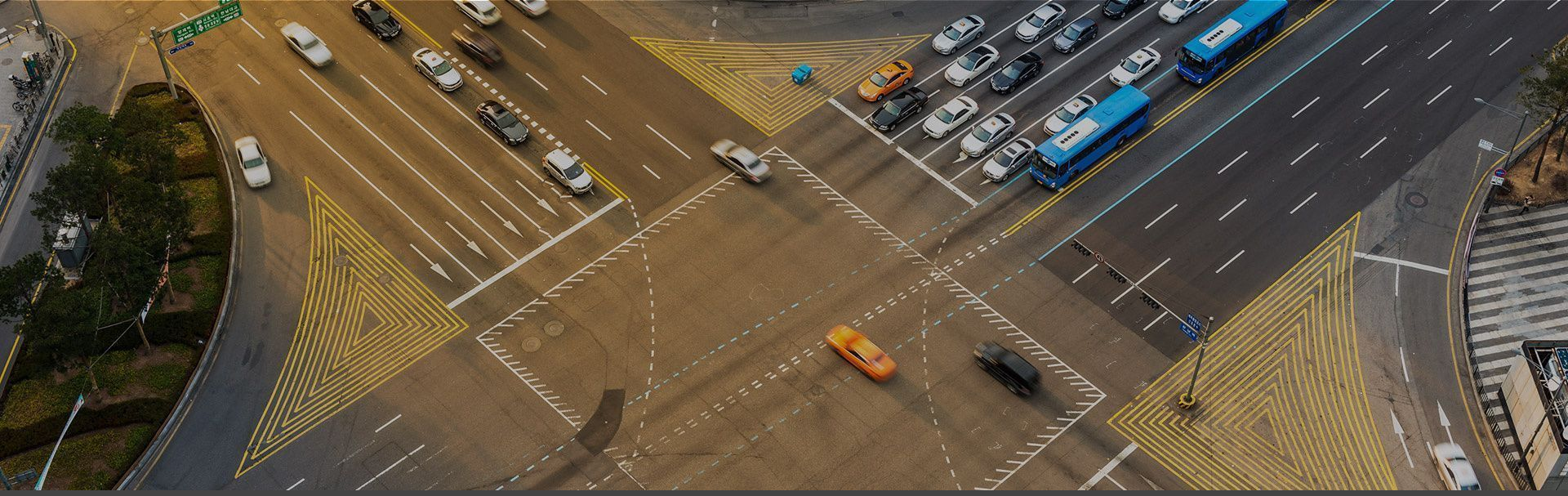
## YAZILIM ÜZERİNE STAJ BAŞVURULARINIZ İÇİN



cubicl

Z Yazılım Ltd. Şti.

- Cubicl İş Takip Programı, Web ve Mobil Uygulama Geliştirme
- <https://cubicl.io>



## ELEKTRONİK ÜZERİNE STAJ BAŞVURULARINIZ İÇİN



ISSD Bilişim Elektronik A.Ş.

- Akıllık Trafik Sistemleri, Gömülü Sistemler, Görüntü İşleme
- <https://issd.com.tr>