

Sayıcılar (Counters)

- Sayıcılar n bitlik bir bilgiyi tutmanın yanısıra her saat çevriminde tuttıkları değeri artıran veya azaltan ardışıl devrelerdir.
- Genel olarak iki gruba ayrılır:
 - Senkron sayıcılar
 - Asenkron sayıcılar (Ripple sayıcılar)
- Yaptığı işe göre oldukça fazla kullanılan sayıcılar standart hale gelmiştir ve hazır devre olarak piyasada bulunmaktadır. Biz bu derste birkaç çeşit sayıcı tasarlayacağız. Ayrıca, senkron ve asenkron sayıcılar arasındaki farkları inceleyeceğiz.

Senkron Sayıcılar

- Senkron sayıcılarda tüm flip-flop'lara uygulanan ortak bir saat (clock) devresi yer almaktadır. Böylece tüm flip-flop'lar senkronize bir şekilde saatin her bir pozitif kenarında tetiklenmekte ve çıkış üretmektedir.
- Örnek:** 0'dan 7'ye kadar sayan bir binary sayıcı tasarlayalım. 000-111 arasında sayacağından 3 bit gerekmektedir. Dolayısıyla da 3 tane flip-flop kullanılacaktır.

Durum Tablosu →

Şimdiki Durum			Gelecek Durum		
Q ²	Q ¹	Q ⁰	Q ²	Q ¹	Q ⁰
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0

J0

		Q_1Q_0			
		1	\emptyset	\emptyset	1
Q_2	1	1	\emptyset	\emptyset	1
	\emptyset	1	\emptyset	\emptyset	1

J1

		Q_1Q_0			
		0	1	\emptyset	\emptyset
Q_2	0	0	1	\emptyset	\emptyset
	1	0	1	\emptyset	\emptyset

J2

		Q_1Q_0			
		0	0	1	0
Q_2	0	0	0	1	0
	1	\emptyset	\emptyset	\emptyset	\emptyset

K0

		Q_1Q_0			
		\emptyset	1	1	\emptyset
Q_2	\emptyset	\emptyset	1	1	\emptyset
	1	\emptyset	1	1	\emptyset

K1

		Q_1Q_0			
		\emptyset	\emptyset	1	0
Q_2	\emptyset	\emptyset	\emptyset	1	0
	1	\emptyset	\emptyset	1	0

K2

		Q_1Q_0			
		\emptyset	\emptyset	\emptyset	\emptyset
Q_2	\emptyset	\emptyset	\emptyset	\emptyset	\emptyset
	1	0	0	1	0

J0

	Q ₁ Q ₀			
Q ₂	1	∅	∅	1
Q ₂	1	∅	∅	1

J1

	Q ₁ Q ₀			
Q ₂	0	1	∅	∅
Q ₂	0	1	∅	∅

J2

	Q ₁ Q ₀			
Q ₂	0	0	1	0
Q ₂	∅	∅	∅	∅

K0

	Q ₁ Q ₀			
Q ₂	∅	1	1	∅
Q ₂	∅	1	1	∅

K1

	Q ₁ Q ₀			
Q ₂	∅	∅	1	0
Q ₂	∅	∅	1	0

K2

	Q ₁ Q ₀			
Q ₂	∅	∅	∅	∅
Q ₂	0	0	1	0

$$J0 = 1$$

$$K0 = 1$$

$$J1 = Q0$$

$$K1 = Q0$$

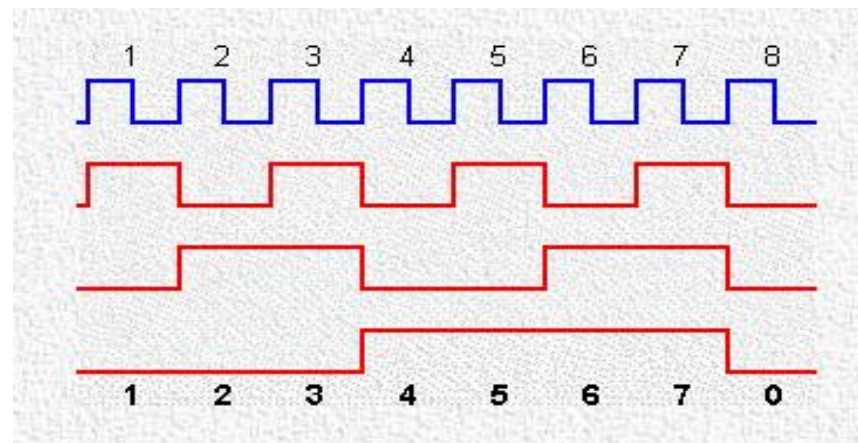
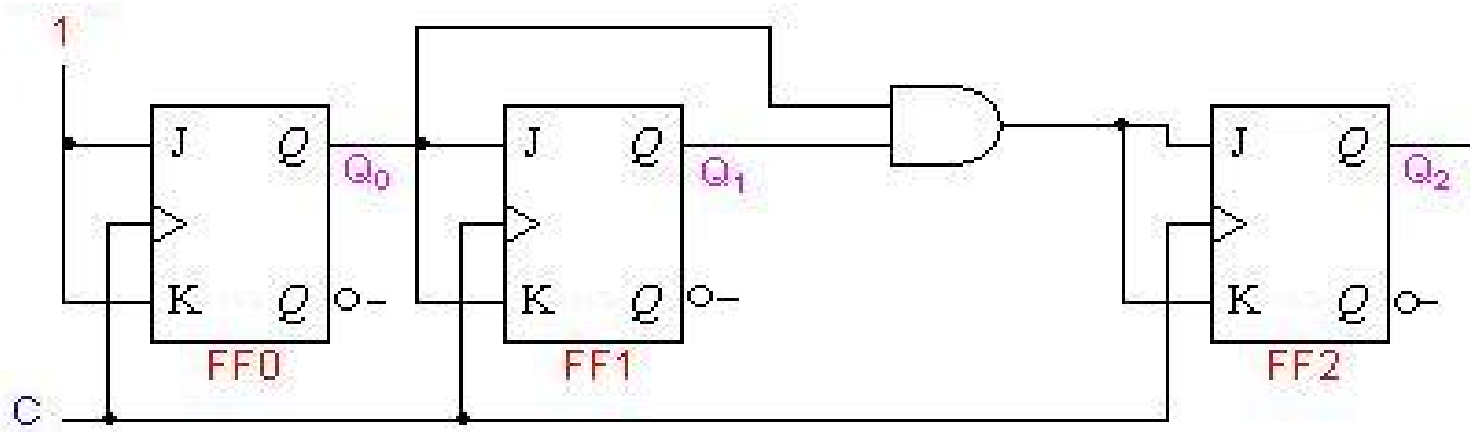
$$J2 = Q1Q0$$

$$K2 = Q1Q0$$

FF0: $J_0 = 1$
 $K_0 = 1$ \rightarrow $Q(t+1) = Q'(t)$

FF1: $J_1 = Q_0$
 $K_1 = Q_0$ \rightarrow $Q_0 = 0$ ise, $J = K = 0$, $Q(t+1) = Q(t)$
 $Q_0 = 1$ ise, $J = K = 1$, $Q(t+1) = Q'(t)$

FF2: $J_2 = Q_1 Q_0$
 $K_2 = Q_1 Q_0$ \rightarrow $Q_0 = 0$ veya $Q_1 = 0$ veya $Q_0 = Q_1 = 0$ ise, $J = K = 0$, $Q(t+1) = Q(t)$
 $Q_0 = Q_1 = 1$ ise, $J = K = 1$, $Q(t+1) = Q'(t)$

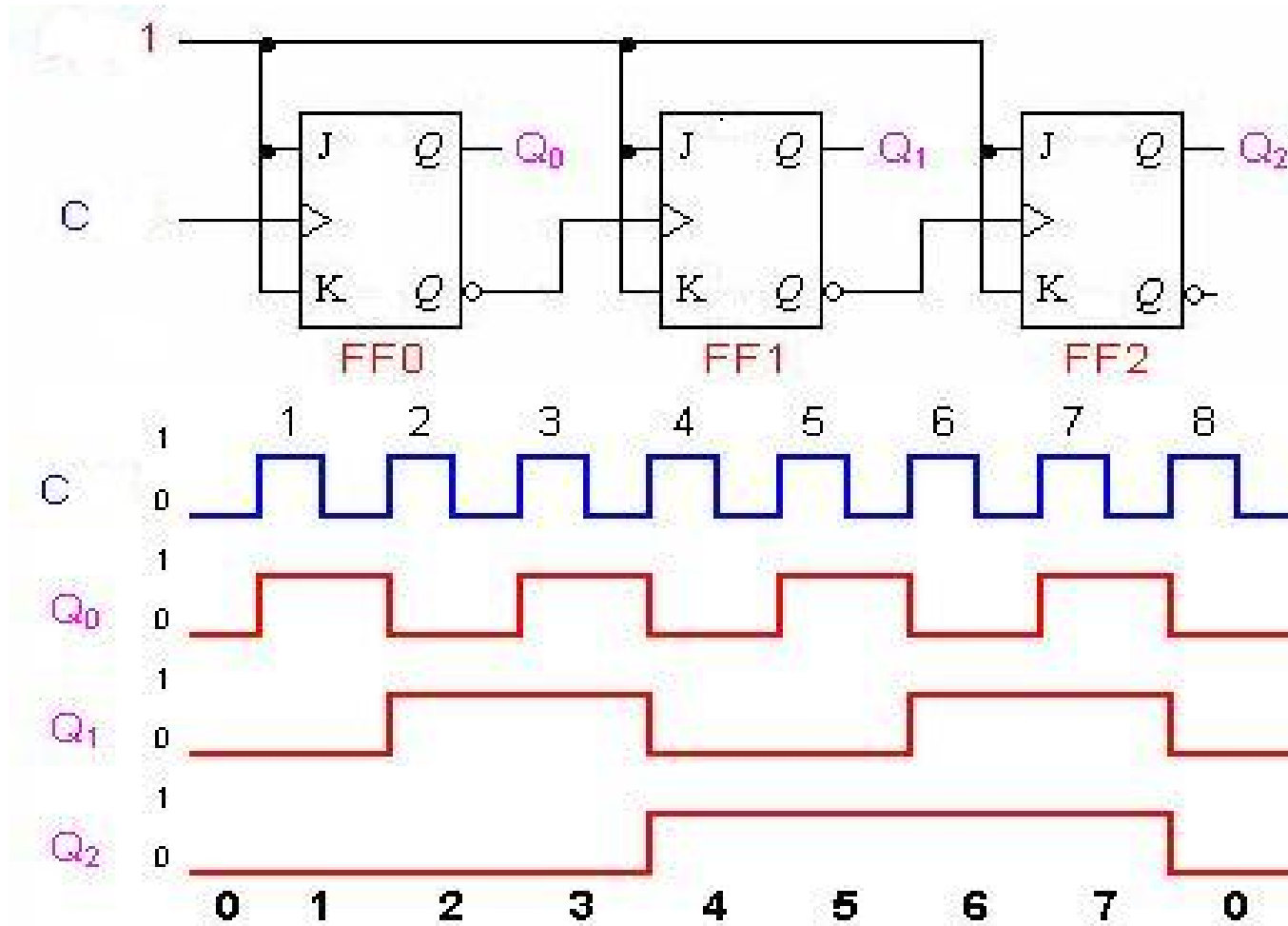


3 bitlik *senkron* binary sayıcı

Asenkron Sayıcılar (Ripple Sayıcılar)

- Senkron sayıcıların sentezi için bildiğimiz ardışıl devre sentezi yöntemi uygulanmasına rağmen asenkron sayıcılar için adım adım izlenebilecek geliştirilmiş bir yöntem yoktur. Asenkron sayıcıların sentezi sezgisel olarak yapılır.
- Senkron sayıcılarda tüm flip-flop'lara uygulanan ortak saat girişi asenkron sayıcılarda sadece sayının en önemsiz bitine karşılık gelen flip-flop'un saat girişine uygulanır. Diğer flip-flop'ların saat girişine ise bir önceki flip-flop'un çıkışının tümleyeni uygulanır.
- Asenkron sayıcıların önemli bir dezavantajı: Hızları ardarda dizilen flip-flop'ların her birinden gelen yayılma zamanı ile sınırlıdır. Senkron sayıcılarda ise hızı saatin periyotunu ayarlayarak biz belirleriz.

• Örnek:



$C1 = Q_0'$

$C2 = Q_1'$

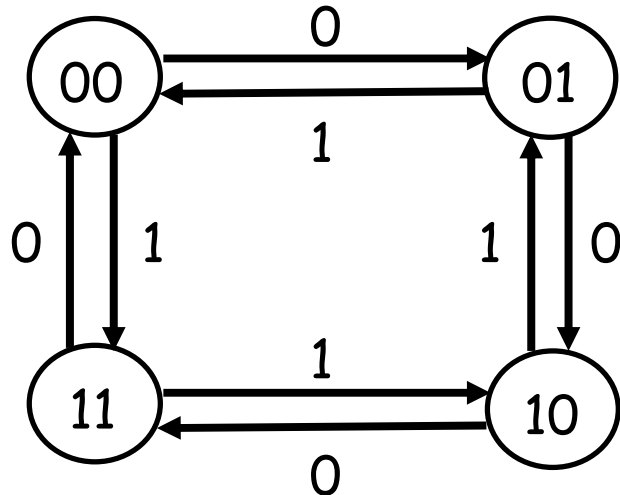
3 bitlik *asen kron* binary sayıcı

Senkron Yukarı/Aşağı Sayıcı

- 2-bitlik senkron binary yukarı/aşağı sayıcı
 - Sayıcı çıkışları 00, 01, 10 ve 11 olacak.
 - 1 tane giriş var: X.
 - > X= 0 ise, sayıcı yukarı doğru sayacak
 - > X= 1 ise, sayıcı aşağı doğru sayacak
- İki tane flip-flop'a ihtiyaç var.

Durum diyagramı ve durum tablosu:

Not: g/? çıkış üretilmiyor...



Şimdiki durum		Giriş	Gelecek durum	
Q ₁	Q ₀	X	Q ₁	Q ₀
0	0	0	0	1
0	0	1	1	1
0	1	0	1	0
0	1	1	0	0
1	0	0	1	1
1	0	1	0	1
1	1	0	0	0
1	1	1	1	0

- Eğer D FF kullanırsak:
(D girişleri doğrudan gelecek duruma eşit olur.)

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

		Q ₀ X	
Q ₁	0	1	0
	1	0	1

$$D_1 = Q_1 \oplus Q_0 \oplus X$$

		Q ₀ X	
Q ₁	1	1	0
	1	1	0

$$D_0 = Q_0'$$

- Eğer JK FF kullanırsak:

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	Gel. Durum		Flip flop girişleri			
Q ₁	Q ₀	X	Q ₁	Q ₀	J ₁	K ₁	J ₀	K ₀
0	0	0	0	1	0	x	1	x
0	0	1	1	1	1	x	1	x
0	1	0	1	0	1	x	x	1
0	1	1	0	0	0	x	x	1
1	0	0	1	1	x	0	1	x
1	0	1	0	1	x	1	1	x
1	1	0	0	0	x	1	x	1
1	1	1	1	0	x	0	x	1

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

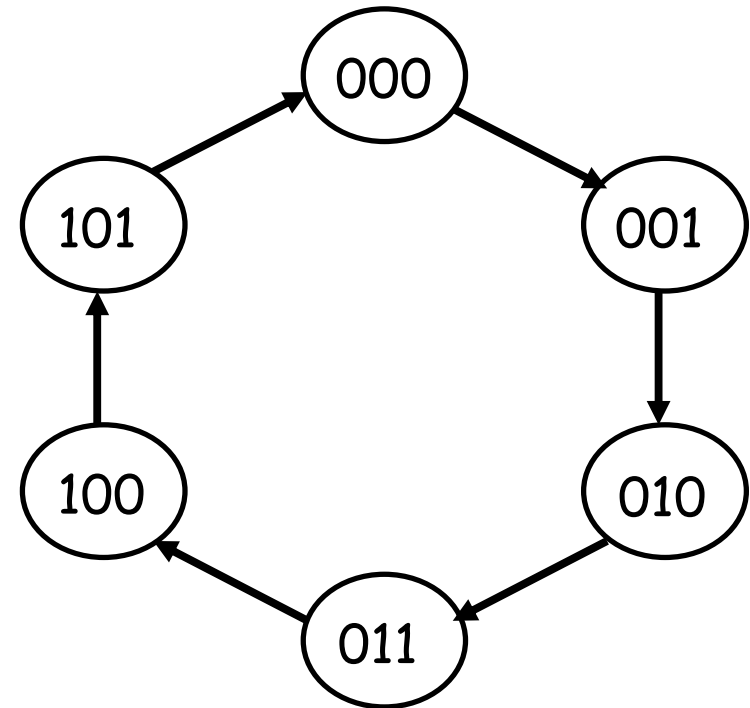
$$J_0 = K_0 = 1$$

0-5 sayan sayıcı için 6 ve 7 ne olacak??

- Eğer 0'dan 5'e sayan bir binary sayıcı tasarlırsak 110 ve 111 durumları ne olacak?

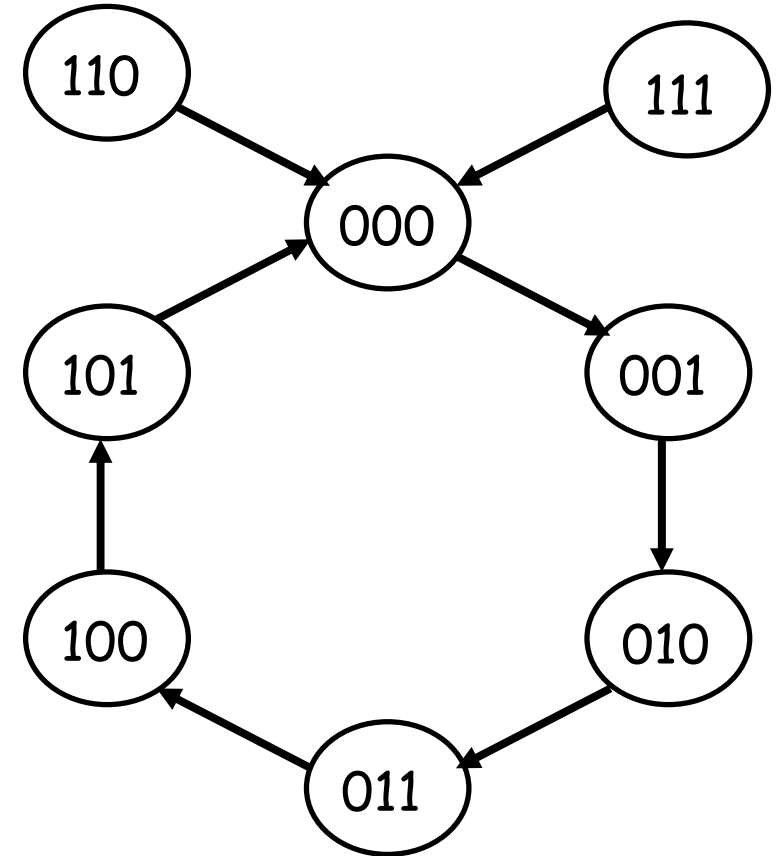
"don't cares **OR** do care..."

Şimdiki Durum			Gelecek Durum		
Q ₂	Q ₁	Q ₀	Q ₂	Q ₁	Q ₀
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	0	0	0
1	1	0	x	x	x
1	1	1	x	x	x



- Daha *güvenilir* bir devre elde etmek için bu kullanılmayan durumlara don't care yerine 0 yazabiliriz.
- Böylece bir şekilde devre bu kullanılmayan durumlara girse bile anlamlı bir sonuç olacaktır.
- Bu bir **kendi-başlayan sayıcı**dır.

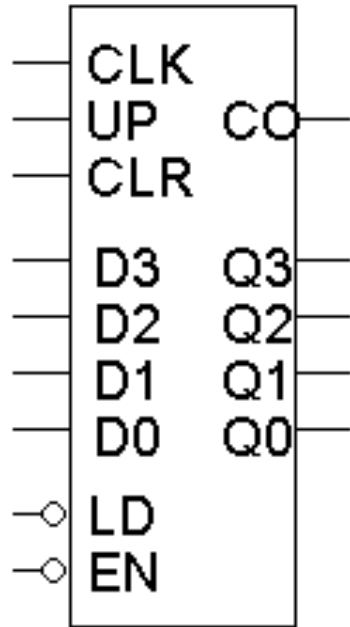
Şimdiki Durum			Gelecek Durum		
Q ₂	Q ₁	Q ₀	Q ₂	Q ₁	Q ₀
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	0	0	0
1	1	0	0	0	0
1	1	1	0	0	0



Daha kapsamlı sayıcılar

- Anlatılanlara göre daha kapsamlı sayıcılar da tasarlanabilir veya standart olarak hazırda bulunabilir:

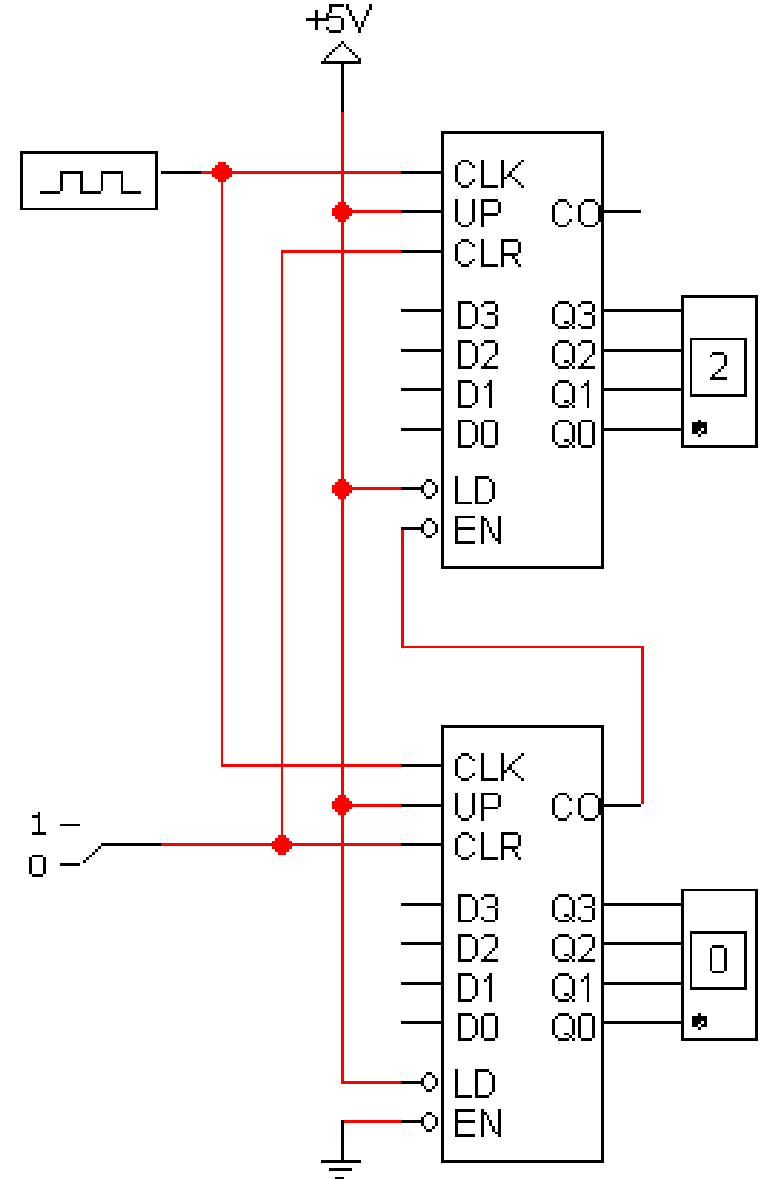
Bir 4 bitlik standart sayıcı:



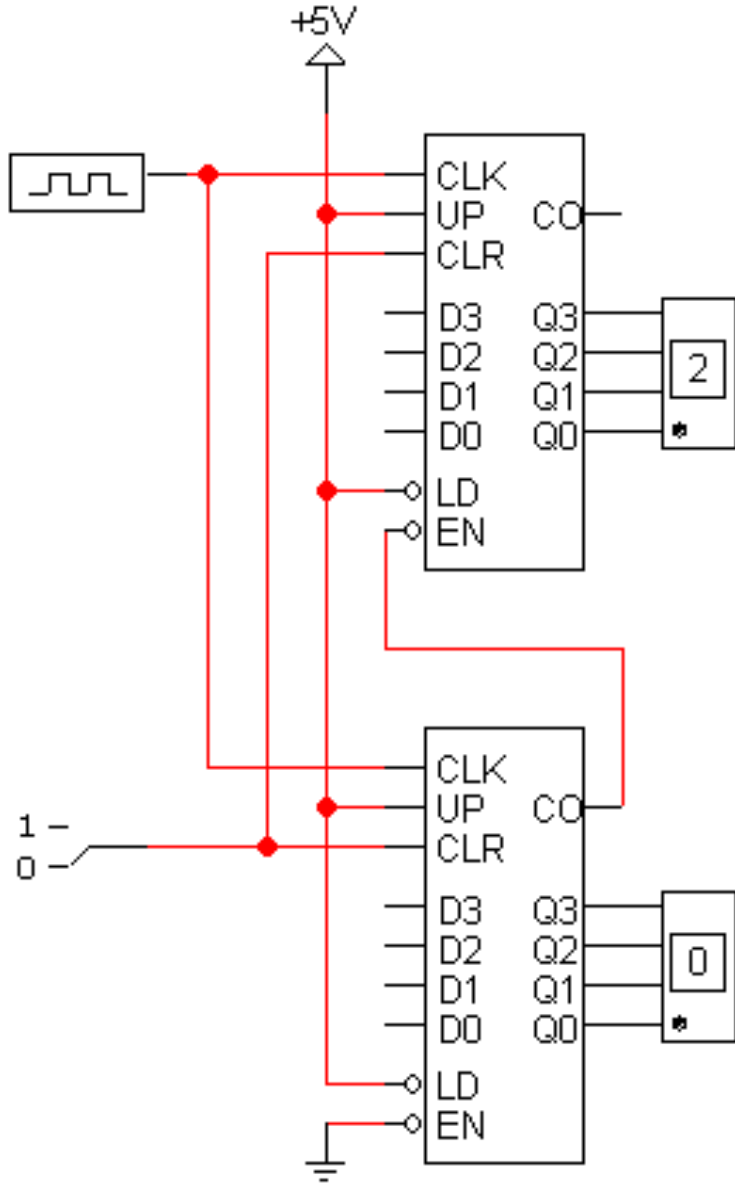
- **UP** girişinin 1 veya 0 olma durumuna göre yukarı veya aşağı sayar.
- **CLR=1** yapılarak sayıcı hemen sıfırlanabilir ($Q_3Q_2Q_1Q_0 = 0000$).
- **D3-D0**'a istenilen 4 bit değer set edilerek ve **LD=0** yapılarak, bir sonraki çıkış belirlenebilir ($Q_3Q_2Q_1Q_0 = D_3D_2D_1D_0$).
- **EN** (Aktif-Düşük (Active-Low)) girişi yardımıyla sayıcı enable veya disable yapılabilir. Sayıcı disable (**EN=1**) olduğunda çıkışındaki sayı aynen korunur (artmaz, azalmaz, yüklenmez veya silinmez).
- **CO** (Counter Out) çıkışı normalde 1'dir. Sayıcı maksimum değere (1111) ulaştığında ise 0 olur.

8-bitlik sayıcı

- Bu iki tane 4-bitlik sayıcıdan oluşturulmuş bir 8-bitlik sayıcıdır.
 - Alttaki sayıcı en önemsiz dört bite aittir. Yukarıdaki ise en önemli dört bite aittir.
 - Aşağıdaki sayıcı 1111'e ulaştığında (yani, $CO = 0$ olduğunda), yukarıdakini 1 saat periyodu için aktif (enable) yapar.
- Not:
 - "Clock" ve "clear" girişleri ortaktır.
 - Hexadecimal göstergeler kullanılır.



8-bitlik sayıcı



- 0000 0000 → CO=1 ve EN=1, FF1 disable
- 0000 0001
- ⋮
- 0000 1110
- 0000 1111 → CO=0 ve EN=0, FF1 enable
- 0001 0000 → CO=1 ve EN=1, FF1 disable
- 0001 0001
- ⋮
- 0001 1110
- 0001 1111 → CO=1 ve EN=1, FF1 disable
- 0010 0000 → CO=0 ve EN=0, FF1 enable
- 0001 0001
- ⋮
- 0001 1110
- 0001 1111
- 0011 0000