

SAYISAL TASARIM

Ege Üniversitesi Ege MYO
Mekatronik Programı

BÖLÜM 1

Sayıcılar

İkili (Binary) Sayma

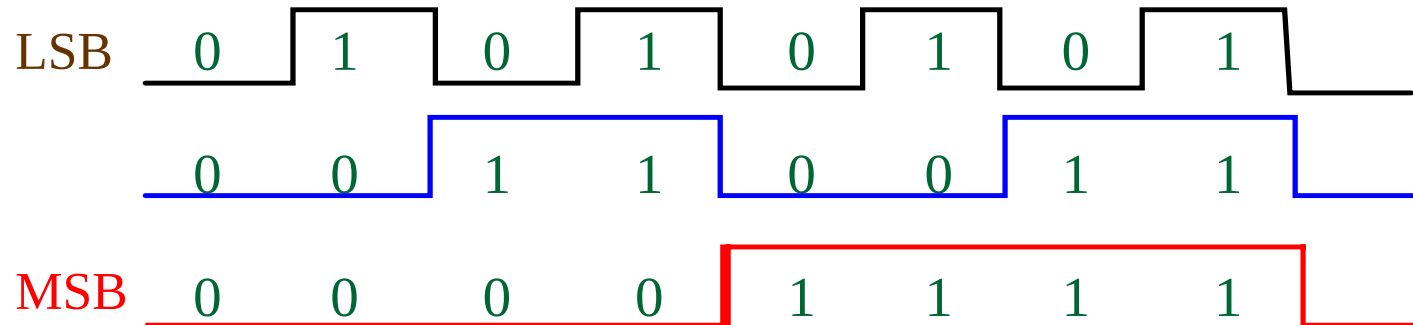
İkili (Binary) sayma 1 ve 0'ların belirli bir düzen içerisinde sıralanması ile yapılır.

Her dört sayıda bir
değişir

0	0	0	LSB her sayma
0	0	1	değerinde değişir.
0	1	0	
0	1	1	Birer atlayarak değişir.
1	0	0	
1	0	1	
1	1	0	
1	1	1	

İkili Sayma

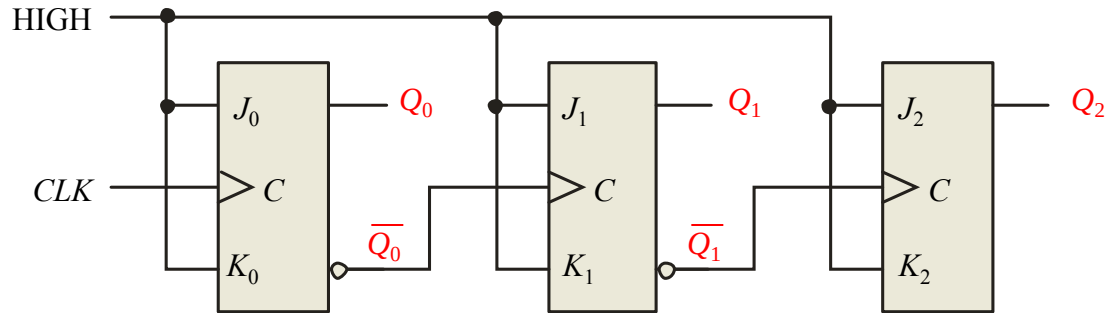
İkili sayma değerlerini mantık seviyesine göre işarete dönüştürebiliriz.



Üç Bit Asenkron Sayıcı

Asenkron sayıcıda saat işareti sadece birinci flip-flopa uygulanır. Diğer flip-floplara bir öncekinin çıkışı işaretleri uygulanır.

J-K flip-floplar toggle modda çalıştırılır.



Üç Bit Asenkron Sayıcı

Q_0 çıkışı saatin yükselen kenarı ile tetiklenir. Bir sonraki FF $\overline{Q_0}$ ile tetiklenir. $\overline{Q_0}$ 'ın yükselen kenarı Q_0 in düşen kenarına eşittir. Sonuçta 3 bit yukarı yönlü sayıcı elde edilir.

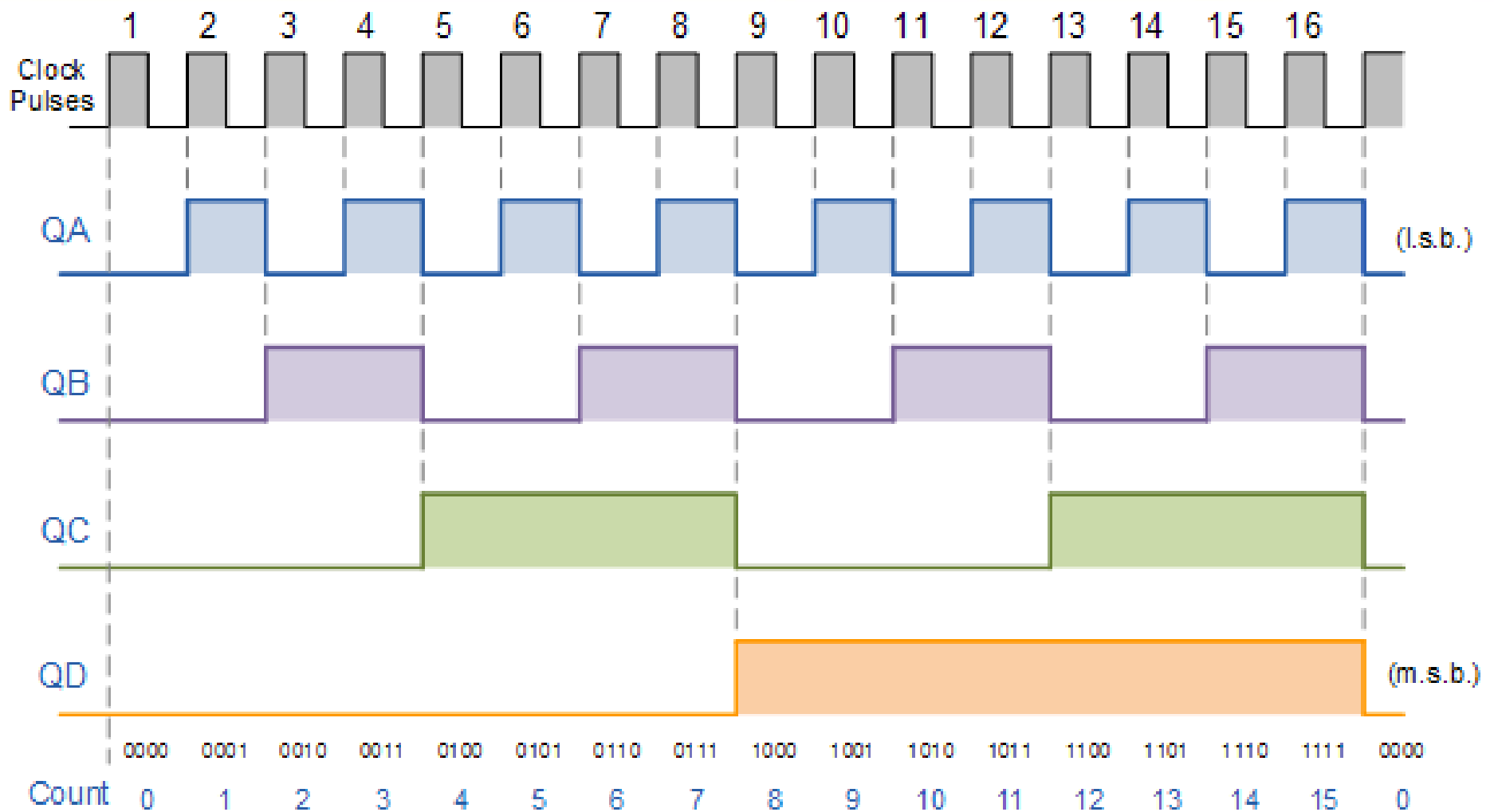
CLK

Q_0

Q_1

Q_2

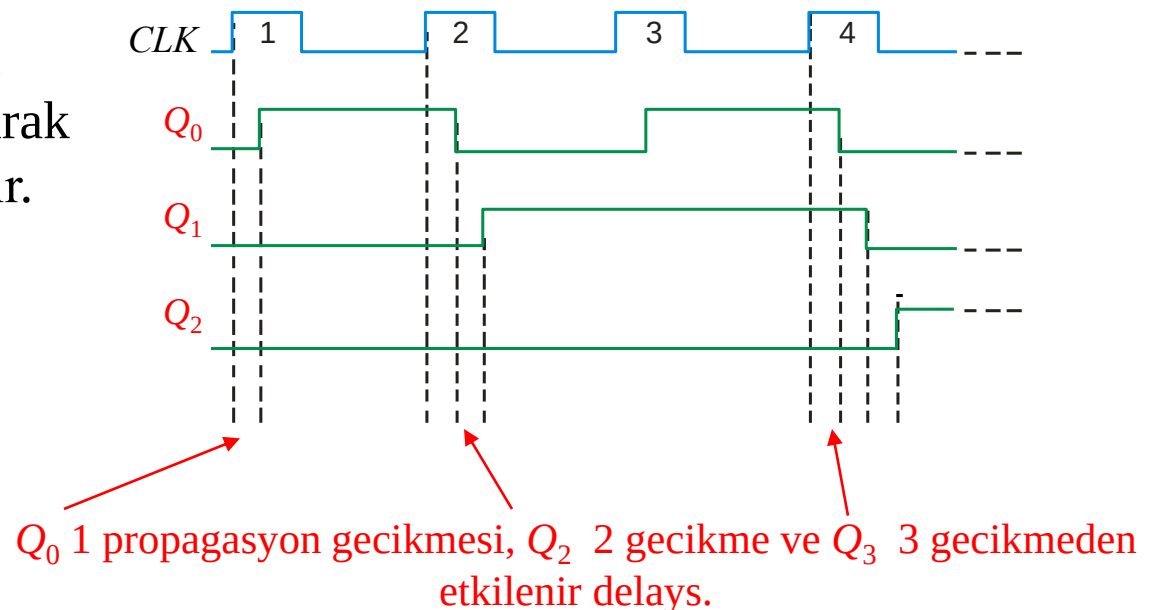
4 Bit Sayıcı



Propagasyon Gecikmesi

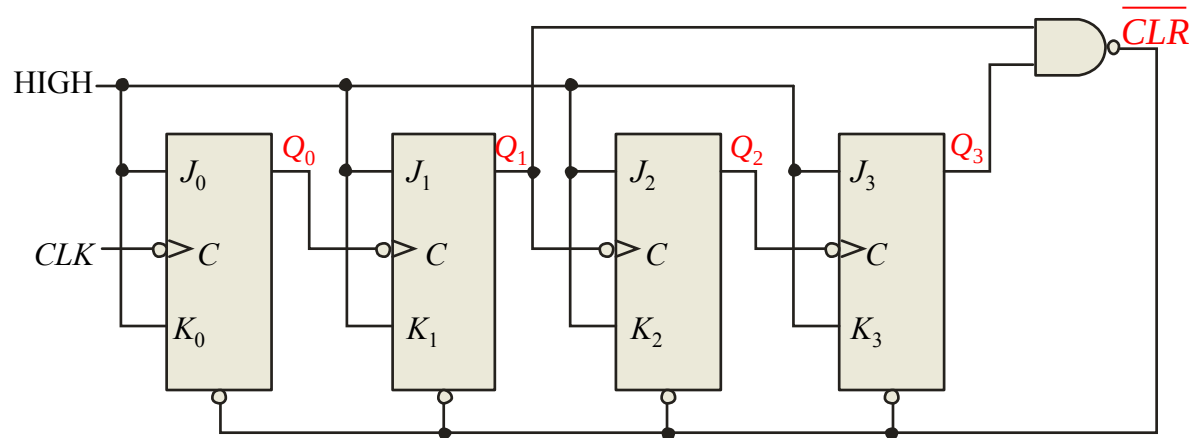
Asenkron sayıcılar aynı zamanda **ripple** sayıcı olarak ta adlandırılır, nedeni ise çıkışların durumları aynı anda değişmez. Yüksek frekanslı tetikleme işareti kullanan sayıcı uygulamalarında bu durum sorun oluşturur.

Her katta oluşan gecikme toplanarak en son kata ulaşır.



Asenkron Onlu Sayıcı

Dört bit asenkron sayıcı 0000 değerinden başlar ve 1111 durumunu saydıktan sonra tekrar 0000 değerine geri döner. Bu sayıcıya HEXADECIMAL sayıcı (onaltılı) adı verilir. Onlu sayıcıda ise 1001 durumundan sonra 0000 durumuna dönülmesi gerekiyor. Bu dönüş için 1010 değeri dekodler ile belirlenip FF'ların asenkron temizleme girişine uygulanması gerekir. CLR girişi kullanılarak sayıcı 0000 durumuna geri döndürülebilir.



Asenkron Onlu Sayıcı

Q_1 ve Q_3 yüksek olduğunda, sayıcıların çıkışları $\overline{CLR}$ hattında oluşan çok dar bir vuru (“glitch”) tarafından temizlenir .

CLK

Q_0

Q_1

Q_2

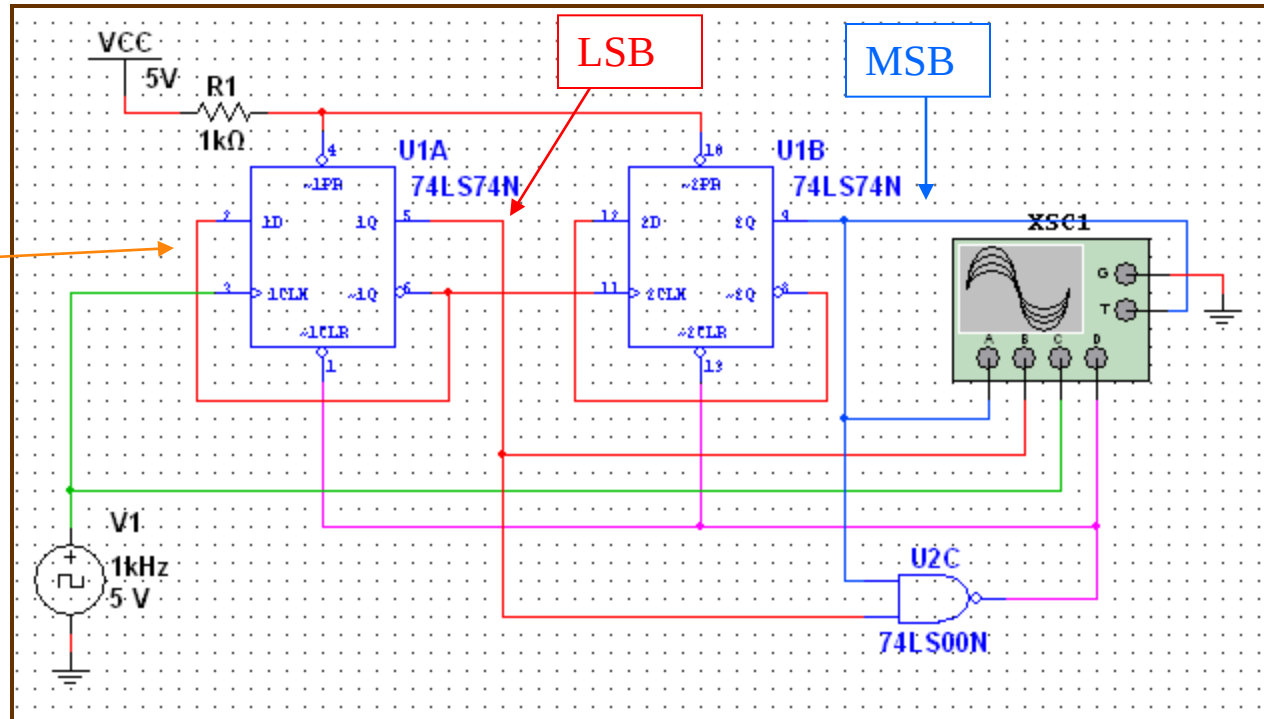
Q_3

$\overline{CLR}$

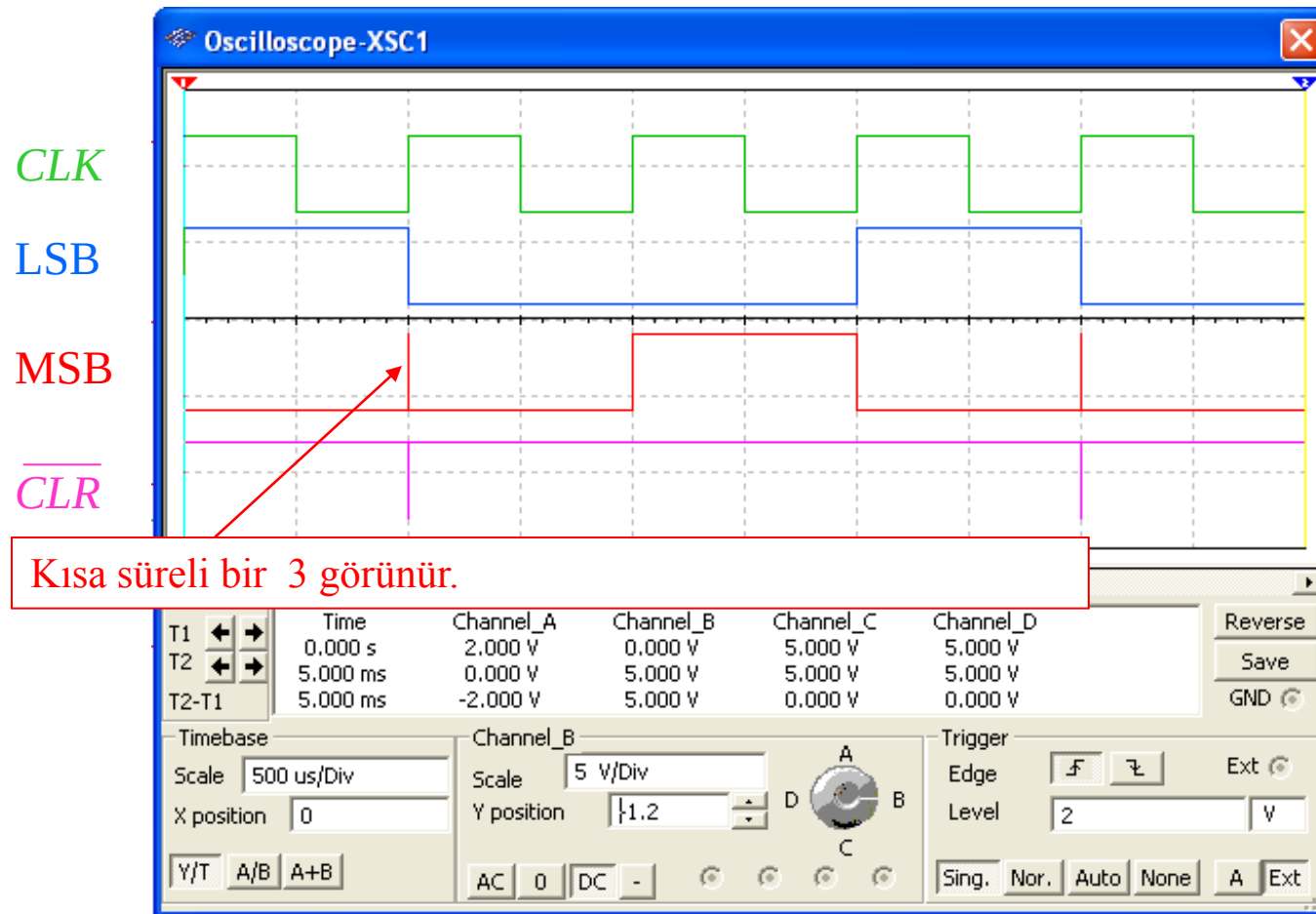
D FF ile Asenkron Sayıcı

D flip-floplar toggle modunda çalışacak şekilde bağlanarak asenkron sayıcı yapımında kullanılabilirler.

*Q 'den D girişine
yapılan bağlantı
toggle moda
çalıştırılır.*



D Flip-Flop ile Asenkron Sayıcı

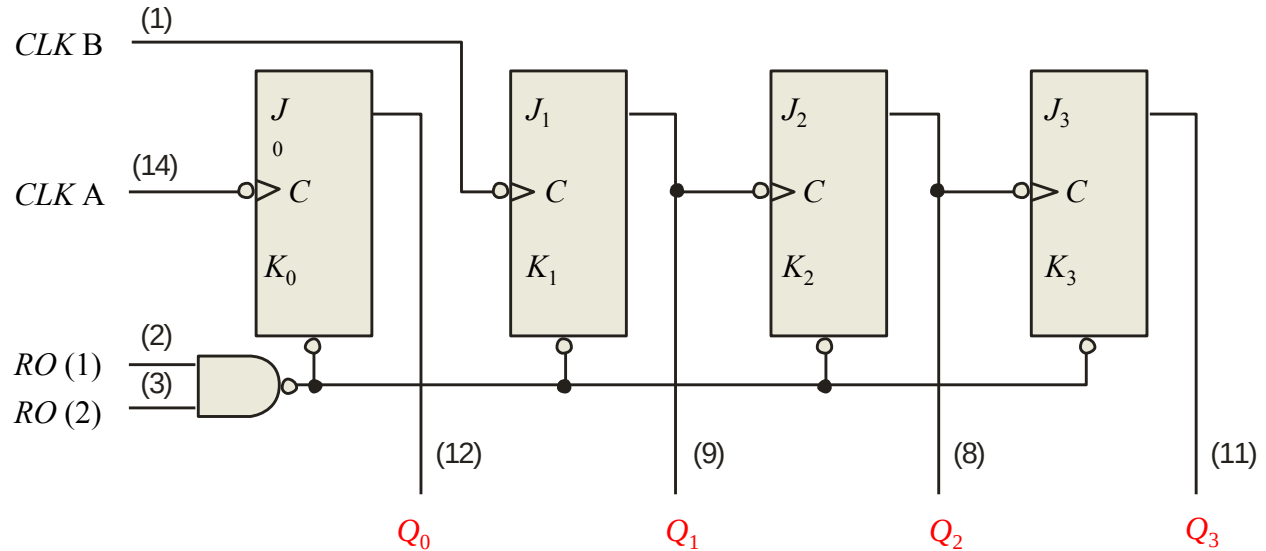


Sayma sırası 0 – 2 – 1 – ($\overline{CLR}$) ...

74LS93A Asenkron Sayıcı

74LS93A içerisinde 4 adet J-K flip-flop bulunur. Birinci FF bağımsız çalışabilir ve *CLK A* girişinden tetiklenebilir. İkinci üçüncü ve dördüncü FF'lar toggle modunda çalışacak şekilde bağlanmıştır ve *CLK B* girişinden tetiklenir.

Sayıcının Q_0 çıkışı *CLK B* girişine bağlanarak 4-bit sayıcı olarak çalıştırılabilir.



Tüm J ve K girişleri içeriden Yüksek seviyeye bağlanmıştır.

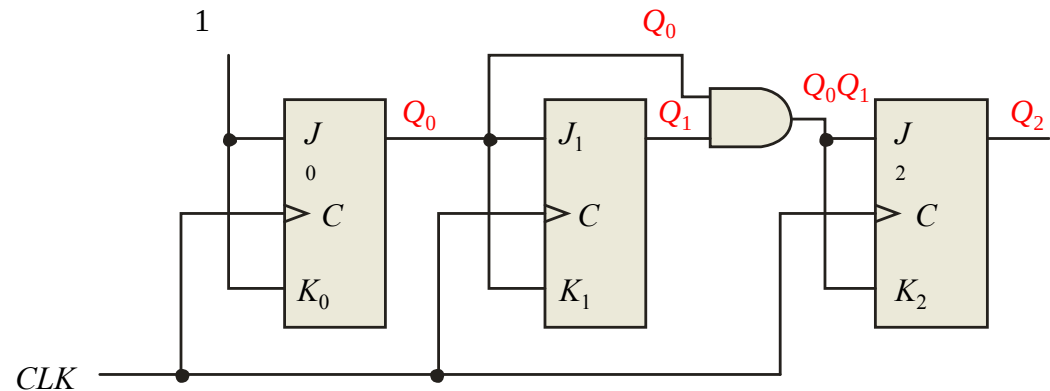
DENEY 1

1. 74LS93A Tüm devresini kullanarak;
 - a. Onlu sayıcı,
 - b. Mod 8 sayıcı,
 - c. Mod 13 sayıcı devrelerini Proteus kullanarak tasarlayın çıkış dalga şekillerini Osilaskopta görüntüleyin.
 - d. Adım 1, 2, ve 3'te tasarladığınız devreleri deney kartınızda kurup çalıştırın. Sayma değerlerini İkili modda çalışırken LED'lerde, Onlu modda çalışırken 7-seg göstergelerde görüntüleyin. (6.3.2014)

Senkron Sayıcılar

Senkron sayıcılarda tüm flip-floplar aynı anda saat işareti ile tetiklenirler. Senkron sayıcılarda propagasyon gecikmelerinin toplanması ile oluşan gecikme ortadan kalkar, buna karşın devrenin yapısı karmaşık hale gelir.

3-bit ikilik senkron sayıcı daha önce öğrendiğimiz 3 bit asenkron sayıcı ile aynı sayma sırasıyla sayar.



Senkron Sayıcılar

Sayıcının çalışmasını anlamak için tablo yöntemini kullanalım

1. Sayma sırasına göre çıkışları yazın ve bu çıkışların oluşması için gerekli girişleri belirleyin.	2. Yeni girişleri kullanarak bir sonraki çıkışlar belirlenir Q_0 toggle, Q_1 , Q_2 aynı kalır.	3. Mevcut çıkışları kullanarak yeni girişleri belirleyin.
---	--	---

Çıkışlar	Girişler					
$Q_2 Q_1 Q_0$	$J_2 = Q_0 Q_1$	$K_2 = Q_0 Q_1$	$J_1 = Q_0$	$K_1 = Q_0$	$J_0 = 1$	$K_0 = 1$
0 0 0	0	0	0	0	1	1
0 0 1	0	0	1	1	1	1
0 1 0	4. Q_2 aynı kalacak Q_1 ve Q_0 terslenecektir.					

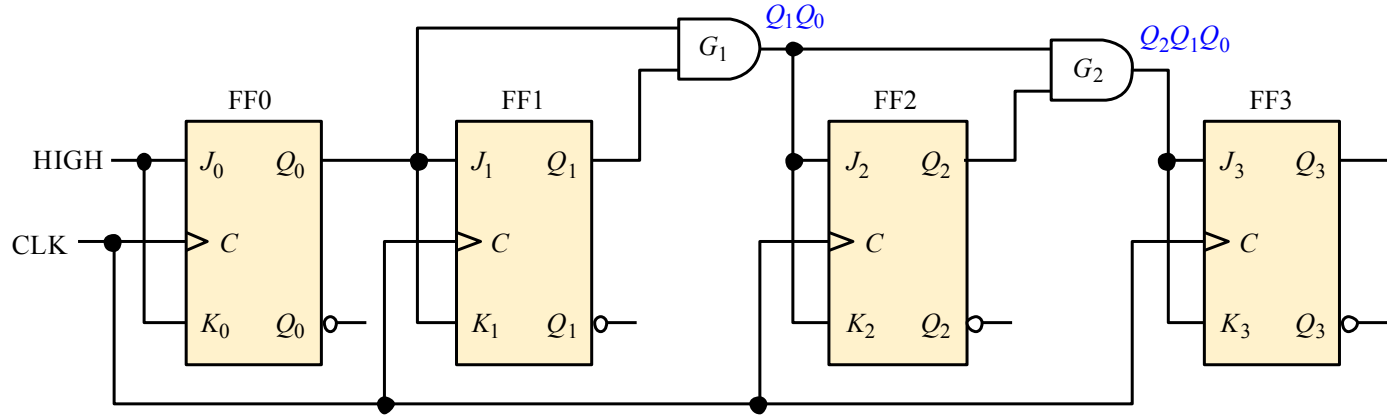
Benzer şekilde tablo tamamlanır.

Senkron Sayıcının Yapısı

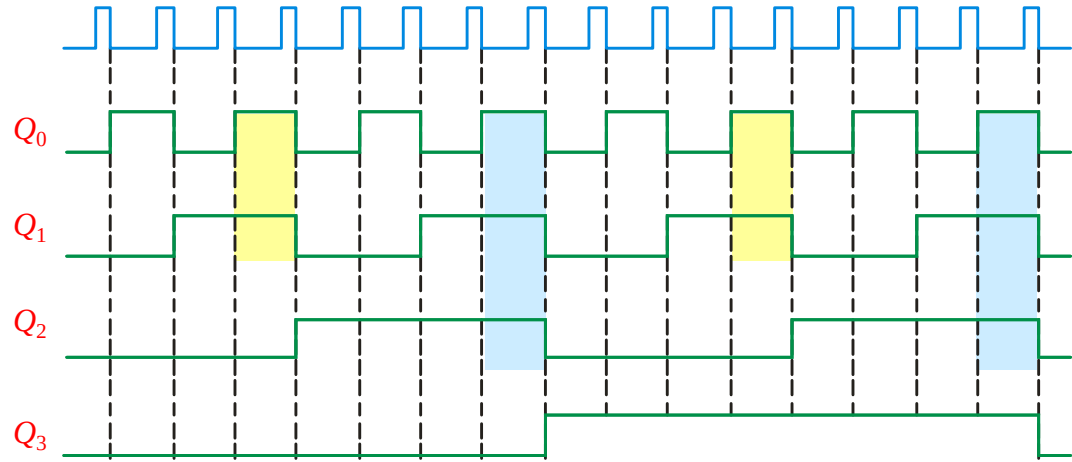
Çıkışlar	Girişlerin mantığı					
$Q_2 Q_1 Q_0$	$J_2 = Q_0 Q_1$	$K_2 = Q_0 Q_1$	$J_1 = Q_0$	$K_1 = Q_0$	$J_0 = 1$	$K_0 = 1$
0 0 0	0	0	0	0	1	1
0 0 1	0	0	1	1	1	1
0 1 0	0	0	0	0	1	1
0 1 1	1	1	1	1	1	1
1 0 0	0	0	0	0	1	1
1 0 1	0	0	1	1	1	1
1 1 0	0	0	0	0	1	1
1 1 1	1	1	1	1	1	1
0 0 0						

Bu durumda sayıcı tüm değerleri saymış ve tekrar başa dönmüştür....

4 bit ikili Senkron Sayıcı



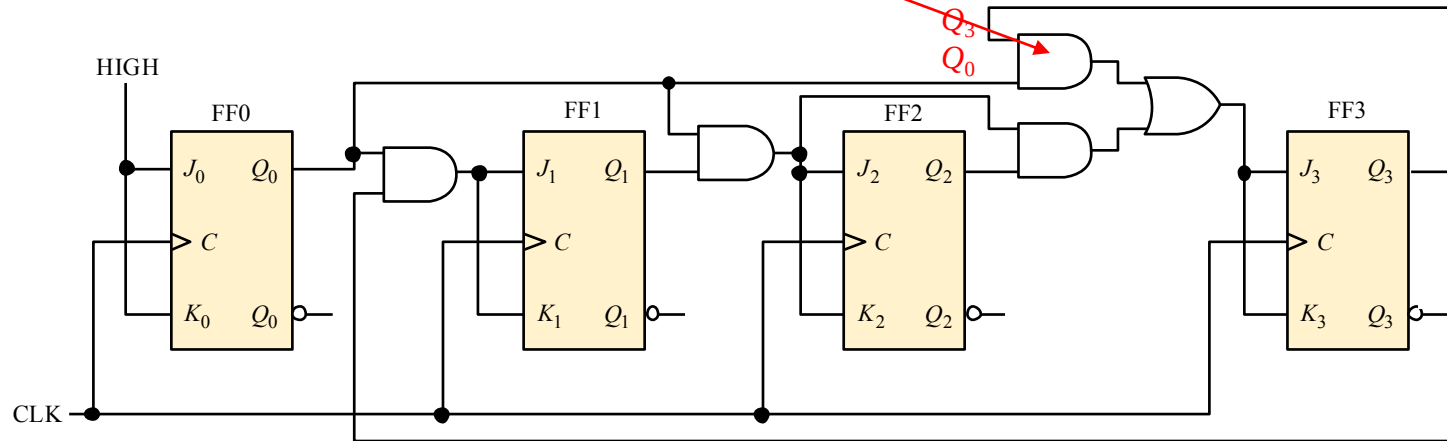
4-bit ikili senkron sayıcıda 3 bit sayıcıya göre bir fazla AND geçidi kullanılır.



BCD Onlu Senkron Sayıcılar

Birkaç ek mantık devre kullanılarak ikili sayıcı BCD sayıcı haline getirilebilir. Sayma değeri 1001 olduktan sonra sayıcı 0000 değerine döner.

Bu geçit sayma değeri 1001, algıladığında FF3 bir sonraki saat vurusunda toggle yapar. FF0 ise her saat vurusunda toggle yapar. Böylece sayma 0000'dan devam eder.



BCD Onlu Senkron Sayıcılar

Onlu sayıcının çıkış dalga şekli:

CLK

Q_0

Q_1

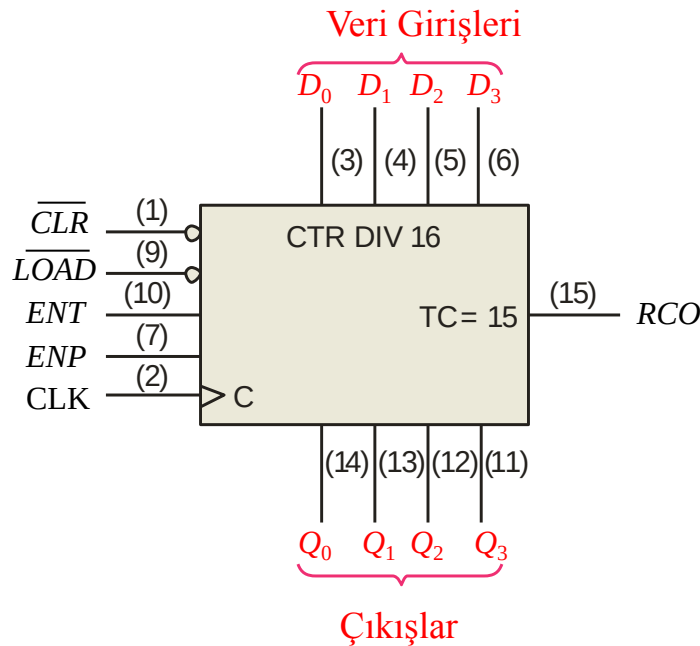
Q_2

Q_3

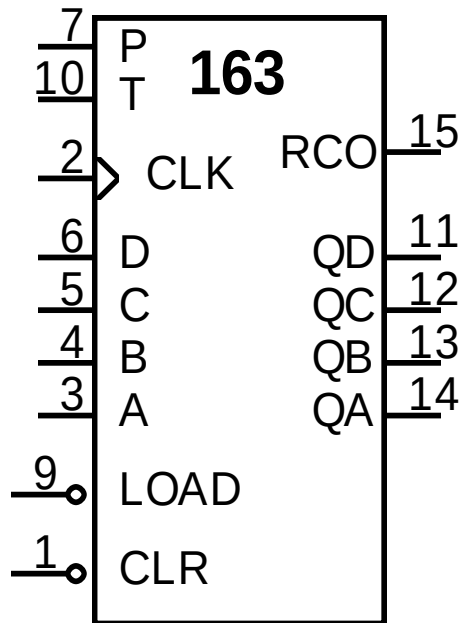
Bu dalga şekli asenkron sayıcı tümdevresi 74LS90 ile de elde edilebilir. aynı tümdevrenin dual versiyonu ise 74LS390 kaskat bağlanabilir. Bu iki asenkron tümdevre senkron sayıcılar kadar hızlı işlemlerde kullanılamazlar. Maksimum çalışma frekansları 35 MHz'dir.

4 bit ikili Senkron Sayıcı

74LS163 basit sayıcının sahip olması gereken ek özelliklere sahip 4-bit Senkron sayıcı tümdevresidir. Paralel başlangıç değeri yükleme girişleri, $\overline{CLR}$ girişi, iki adet seçme girişi ve ripple sayma çıkışı bu tümdevreyi birçok uygulama için kullanışlı hale getirir.



74LS163 Senkron 4-bit ikilik sayıcı



LOAD: Senkron Yükleme girişi

CLR: Temizleme girişi

CLK: Pozitif Kenar Tetiklemeli FF'lar

D, C, B, A: Paralel Veri Yükleme Girişleri

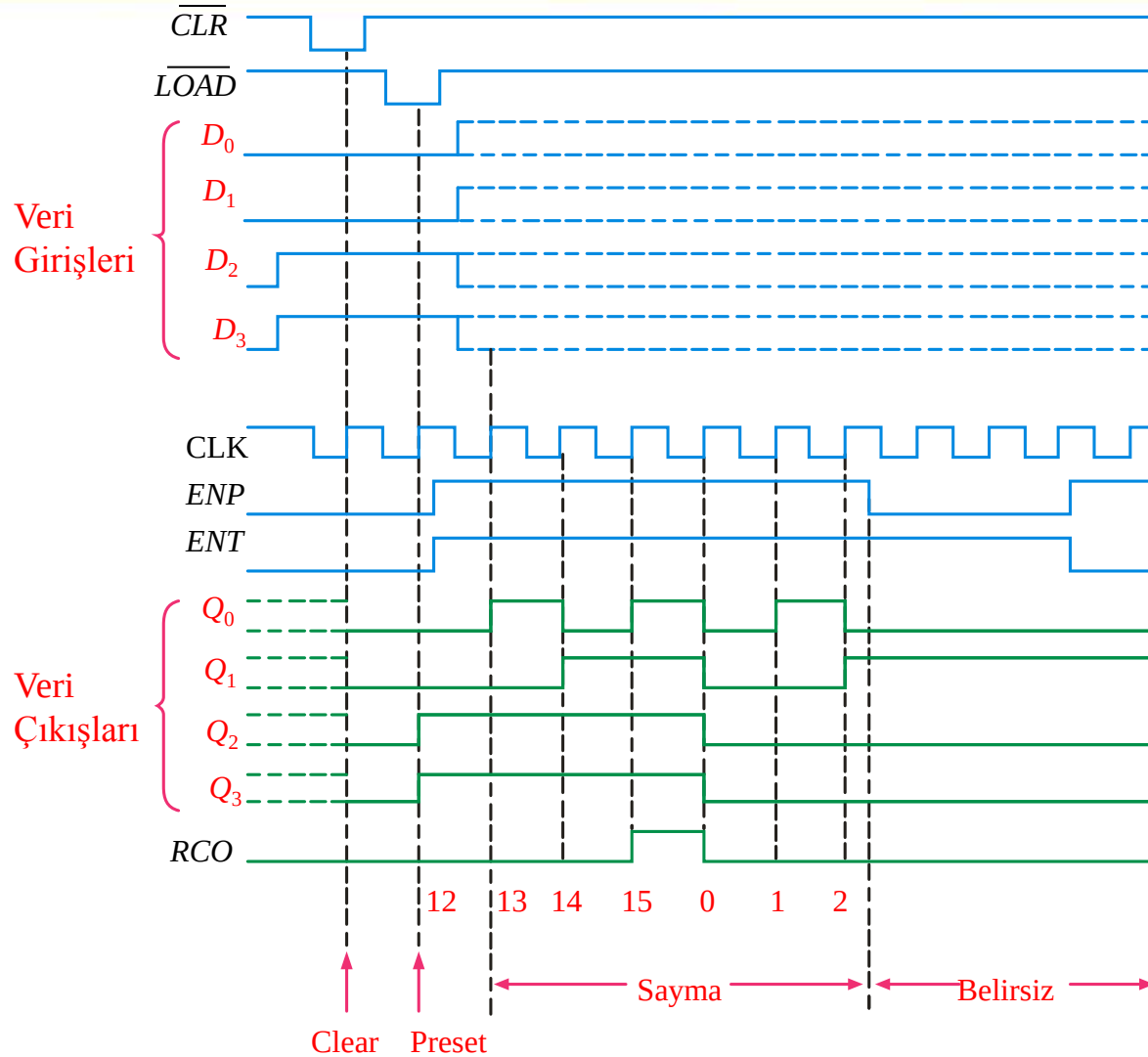
ENP, ENT: (Enable) İzin girişleri: sayıcının sayabilmesi için Her ikisi de "1" olmalıdır.

RCO: Sayıcı en yüksek değerine (1111) geldiğinde çıkış verir (mantık 1), sayıcıların kaskat bağlantısı için kullanılır.

RCO "Ripple Carry Output"

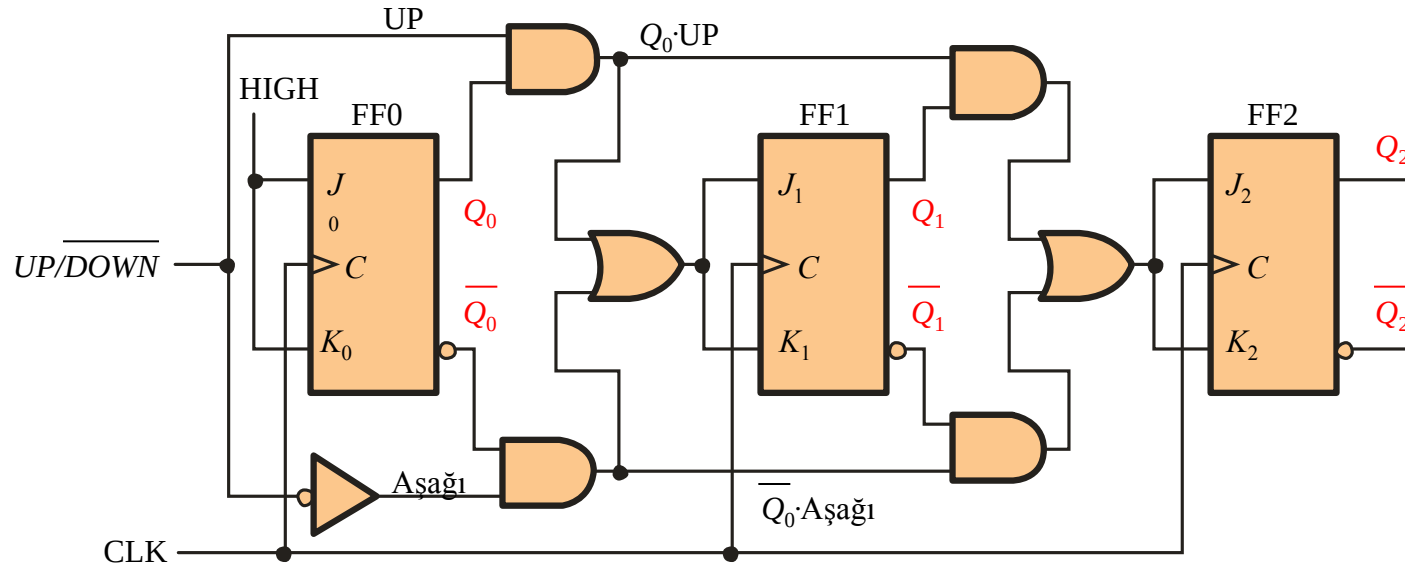
74161: işlevsel olarak aynı, asenkron yükleme ve reset

4 bit ikili Senkron Sayıcı

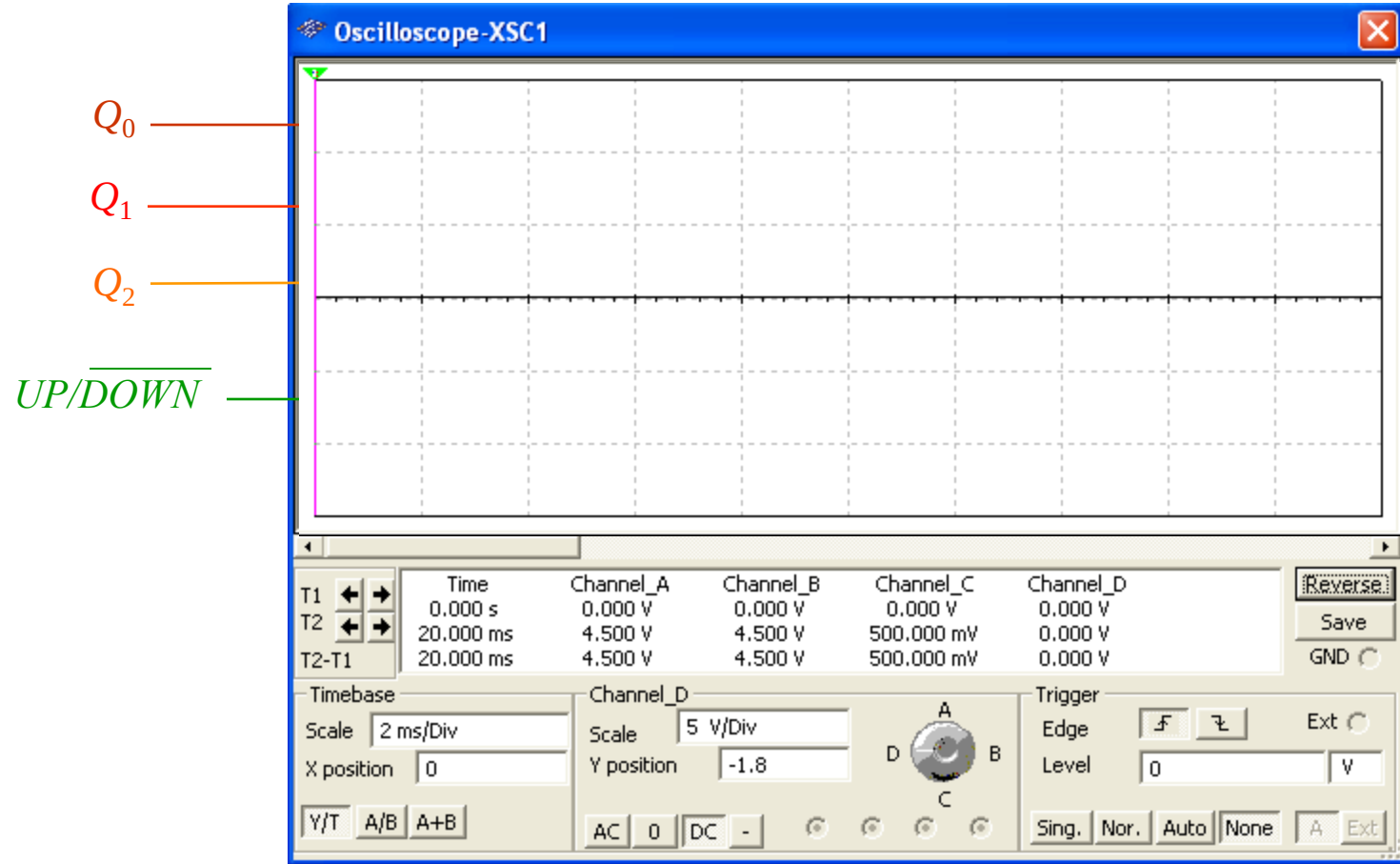


Yukarı/Aşağı (Up/Down) Senkron Sayıcılar

Yukarı/aşağı sayıcılar denetim işaretine bağlı olarak her iki yöne doğru sayma yapabilirler.



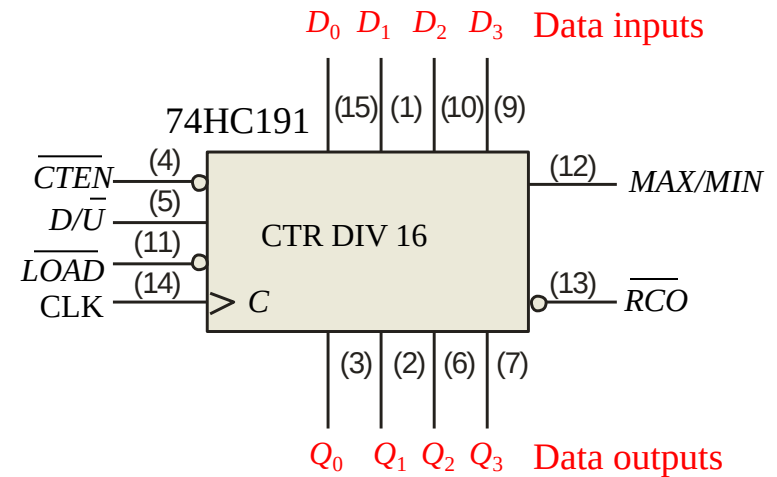
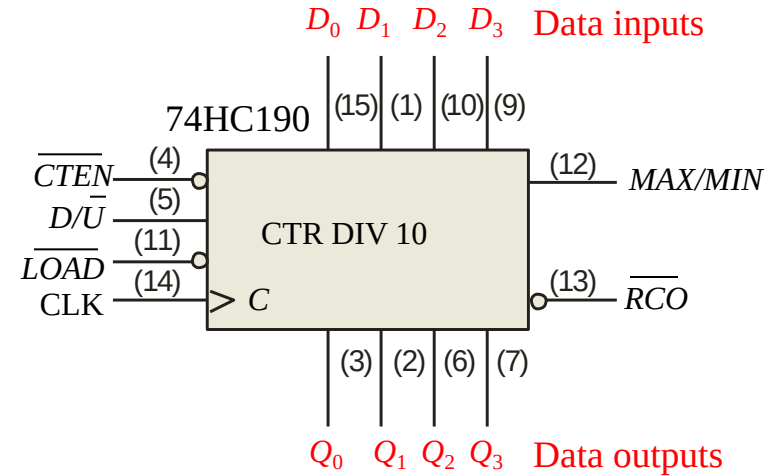
Yukarı/Aşağı (Up/Down) Senkron Sayıcılar



Yukarı/Aşağı (Up/Down) Senkron Sayıcılar

74HC190 yüksek hızlı CMOS senkron yukarı aşağı onlu sayıcıdır. Paralel olarak başlangıç değeri yükleme girişleri, aktif düşük ripple saat çıkışı ($\overline{RCO}$) ve terminal sayma çıkışı MAX/MIN tümdevre üzerinde yer alır.

74HC191 aynı giriş ve çıkışlara sahiptir tek farkı bu tümdevre senkron ikili sayıcıdır.



Yukarı/Aşağı (Up/Down) Senkron Sayıcılar

74HC193 yüksek hızlı CMOS senkron yukarı aşağı ikili sayıcıdır. Paralel olarak başlangıç değeri yükleme girişleri, aktif düşük ripple saat çıkışı (*RCO*) ve terminal sayma çıkışı *MAX/MIN* tümdevre üzerinde yer alır. 191m2den farkı yukarı ve aşağı sayma için ayrı saat girişleri olmasıdır.

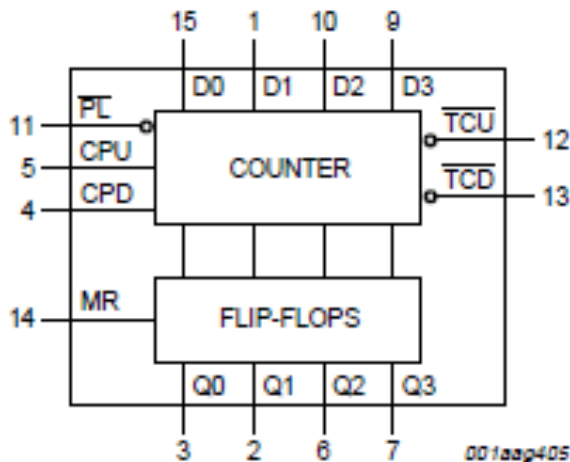
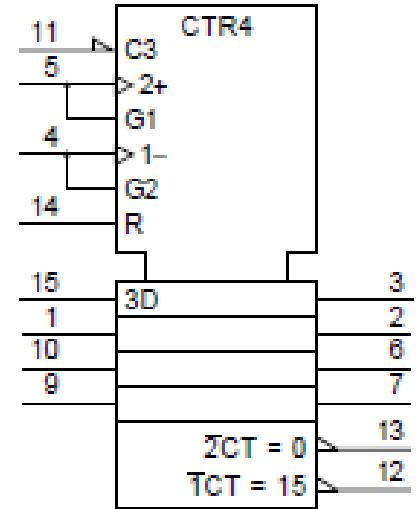


Fig 1. Functional diagram

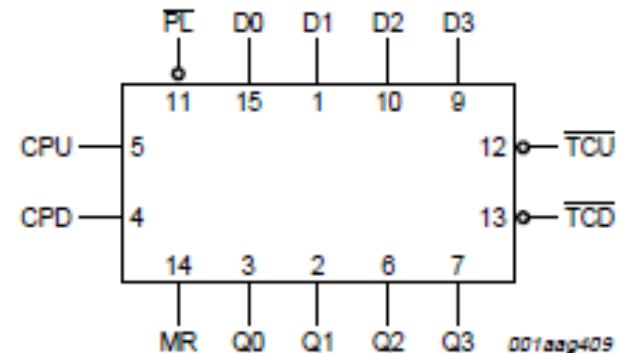


Fig 2. Logic symbol

74HC193

6. Functional description

Table 3. Function table^[1]

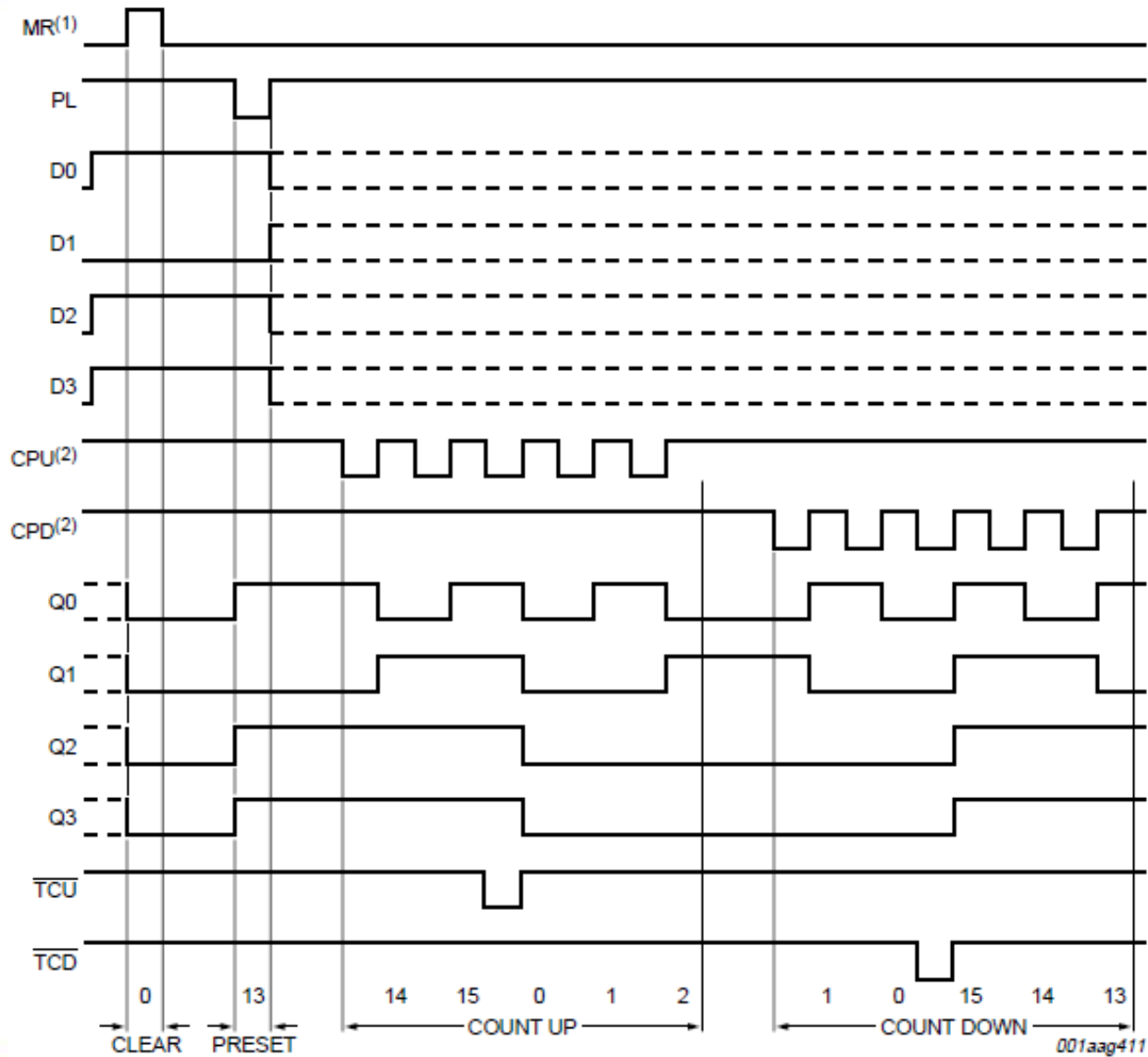
Operating mode	Inputs								Outputs					
	MR	$\overline{PL}$	CPU	CPD	D0	D1	D2	D3	Q0	Q1	Q2	Q3	$\overline{TCU}$	$\overline{TCD}$
Reset (clear)	H	X	X	L	X	X	X	X	L	L	L	L	H	L
	H	X	X	H	X	X	X	X	L	L	L	L	H	H
Parallel load	L	L	X	L	L	L	L	L	L	L	L	L	H	L
	L	L	X	H	L	L	L	L	L	L	L	L	H	H
	L	L	L	X	H	H	H	H	H	H	H	H	L	H
	L	L	H	X	H	H	H	H	H	H	H	H	H	H
Count up	L	H	↑	H	X	X	X	X	count up				H ^[2]	H
Count down	L	H	H	↑	X	X	X	X	count down				H	H ^[3]

- [1] H = HIGH voltage level
 L = LOW voltage level
 X = don't care
 ↑ = LOW-to-HIGH clock transition.

- [2] $\overline{TCU}$ = CPU at terminal count up (HHHH)

- [3] $\overline{TCD}$ = CPD at terminal count down (LLLL).

74HC193 Zamanlama Diyagramı

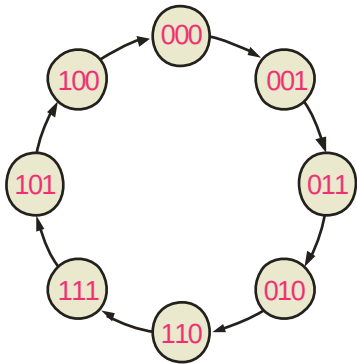


Senkron Sayıcı Tasarımı

Mevcut senkron sayıcı tümdevreleri bir çok gereksinimi karşılarken özel sayma sırası gerektiren uygulamalar için senkron sayıcının FF kullanarak tasarlanması gerekir.

Tasarıma sayma durum diyagramı çizilerek başlanır. Gray kodunda sayıcının durum diyagramı ve geçiş tablosu gösterilmiştir.

Durum Diyagramı:



Sonraki durum Tablosu:

Present State			Next State		
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0
0	0	0	0	0	1
0	0	1	0	1	1
0	1	1	0	1	0
0	1	0	1	1	0
1	1	0	1	1	1
1	1	1	1	0	1
1	0	1	1	0	0
1	0	0	0	0	0

Output Transitions		Flip-Flop Inputs	
Q_N	Q_{N+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

J-K geçiş tablosu sol tarafta olası şimdiki (Q_N) ve gelecek durumları (Q_{N+1}) ve sağ tarafta ise sol taraftaki geçişi sağlayacak girişler yer alır.

The values of J_0 and K_0 required to produce the transition are placed on each map in the present-state cell.

The values of J_0 and K_0 required to produce the transition are placed on each map in the present-state cell.

J_0 map

		Q_0	
		0	1
Q_2Q_1	00	1	
	01		
	11		
	10		X

K_0 map

		Q_0	
		0	1
Q_2Q_1	00	X	
	01		
	11		
	10		1

Output Transitions		Flip-Flop Inputs	
Q_N	Q_{N+1}	J	K
0	→ 0	0	X
0	→ 1	1	X
1	→ 0	X	1
1	→ 1	X	0

Flip-flop transition table

Present State			Next State		
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0
0	0	0	0	0	1
0	0	1	0	1	1
0	1	1	0	1	0
0	1	0	1	1	0
1	1	0	1	1	1
1	1	1	1	0	1
1	0	1	1	0	0
1	0	0	0	0	0

Next-state table

For the present state 000, Q_0 makes a transition from 0 to 1 to the next state.

For the present state 101, Q_0 makes a transition from 1 to 0 to the next state.

$Q_2Q_1 \backslash Q_0$		0	1
00		0	0
01		1	0
11		X	X
10		X	X

$Q_1\bar{Q}_0$

J_2 map

$Q_2Q_1 \backslash Q_0$		0	1
00		0	1
01		X	X
11		X	X
10		0	0

$\bar{Q}_2Q_0$

J_1 map

$Q_2Q_1 \backslash Q_0$		0	1
00		1	X
01		0	X
11		1	X
10		0	X

$\bar{Q}_2Q_0$

Q_2Q_0

J_0 map

$Q_2Q_1 \backslash Q_0$		0	1
00		X	X
01		X	X
11		0	0
10		1	0

$\bar{Q}_1\bar{Q}_0$

K_2 map

$Q_2Q_1 \backslash Q_0$		0	1
00		X	X
01		0	0
11		0	1
10		X	X

Q_2Q_0

K_1 map

$Q_2Q_1 \backslash Q_0$		0	1
00		X	0
01		X	1
11		X	0
10		X	1

$\bar{Q}_2Q_0$

Q_2Q_0

K_0 map

$$J_0 = Q_2 Q_1 + \overline{Q_2} \overline{Q_1} = \overline{Q_2 \oplus Q_1}$$

$$K_0 = Q_2 \overline{Q_1} + \overline{Q_2} Q_1 = Q_2 \oplus Q_1$$

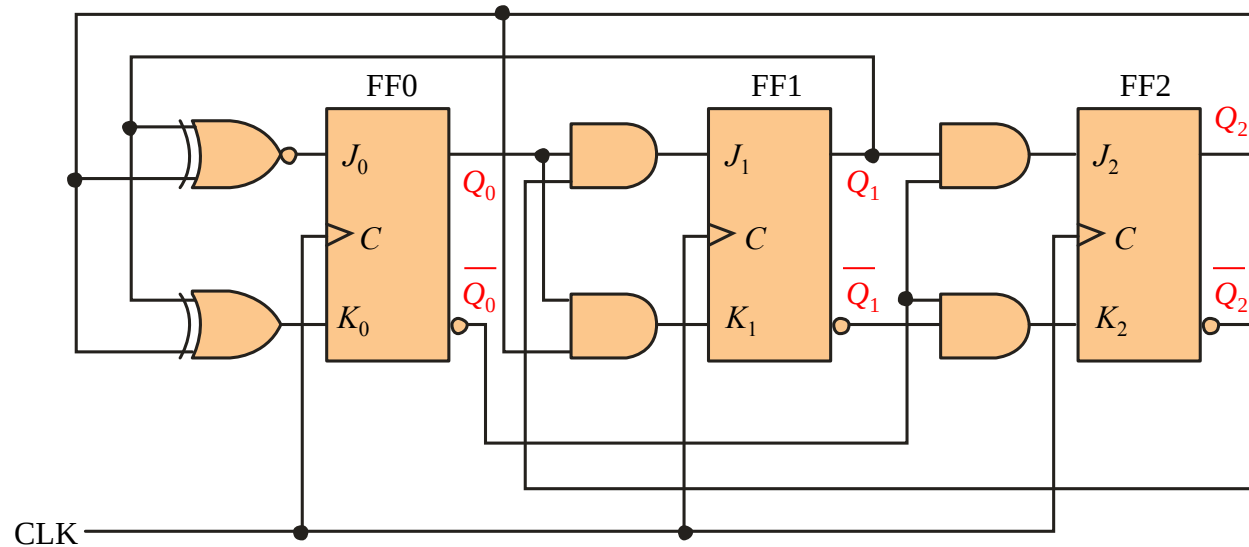
$$J_1 = \overline{Q_2} Q_0$$

$$K_1 = Q_2 Q_0$$

$$J_2 = Q_1 \overline{Q_0}$$

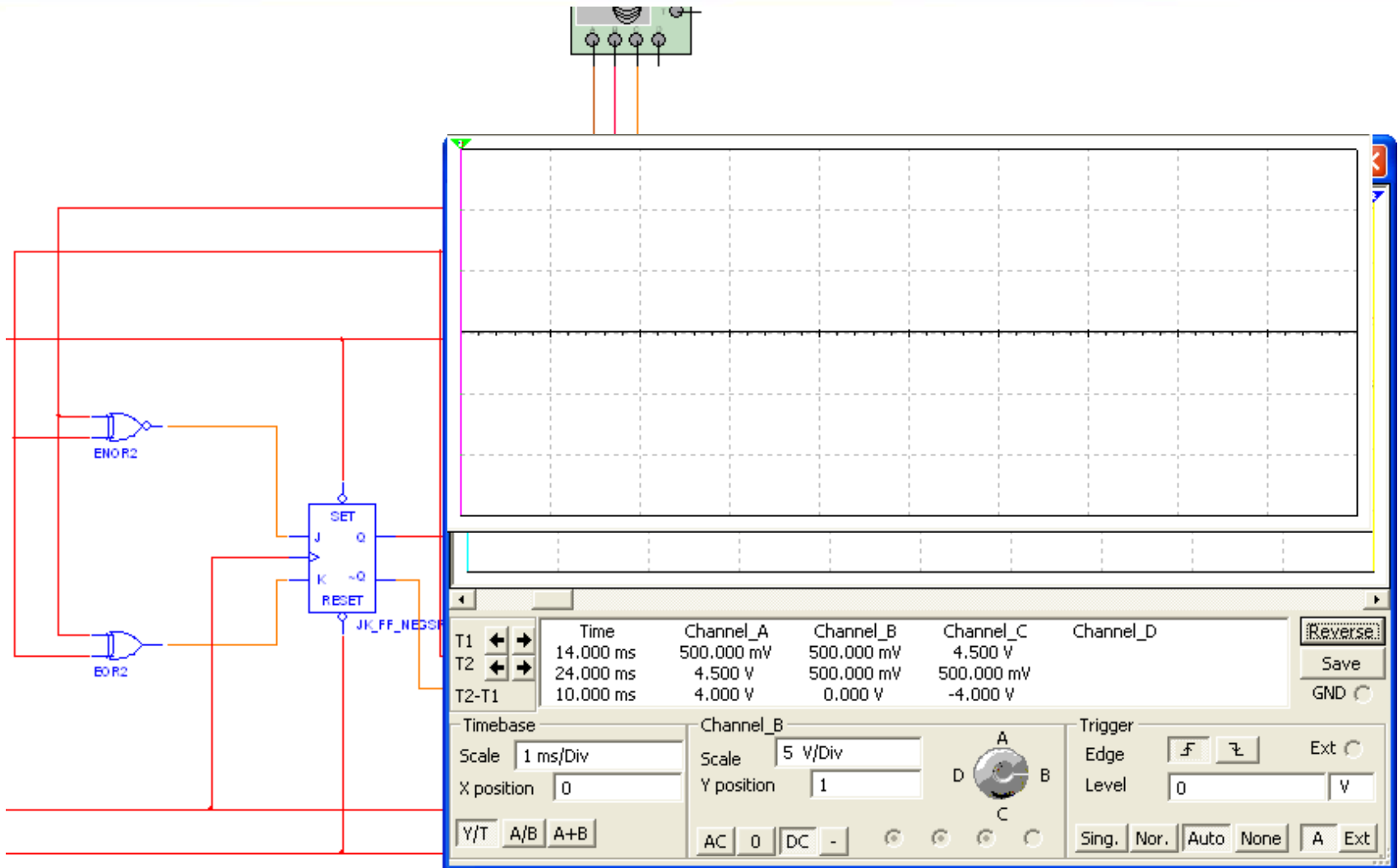
$$K_2 = \overline{Q_1} \overline{Q_0}$$

Senkron Sayıcı Tasarımı

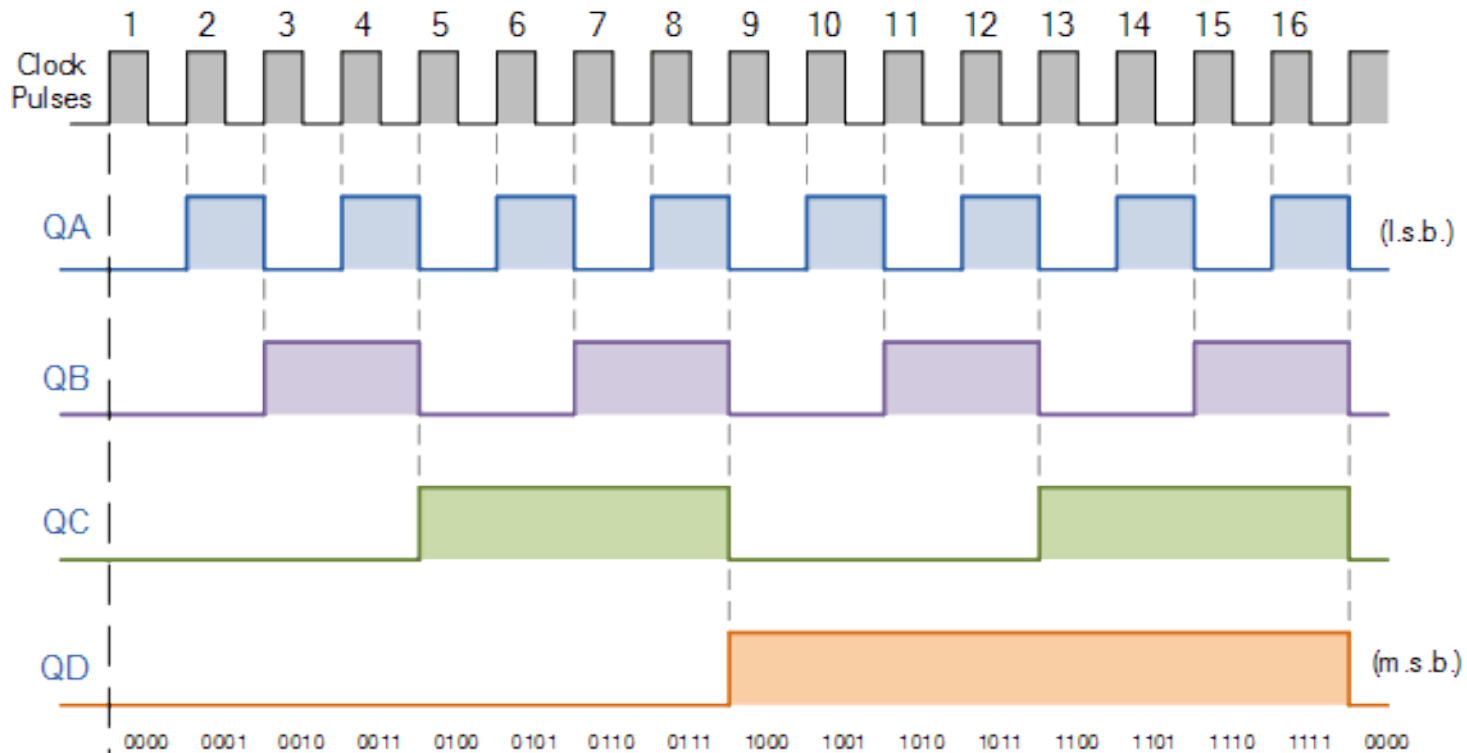
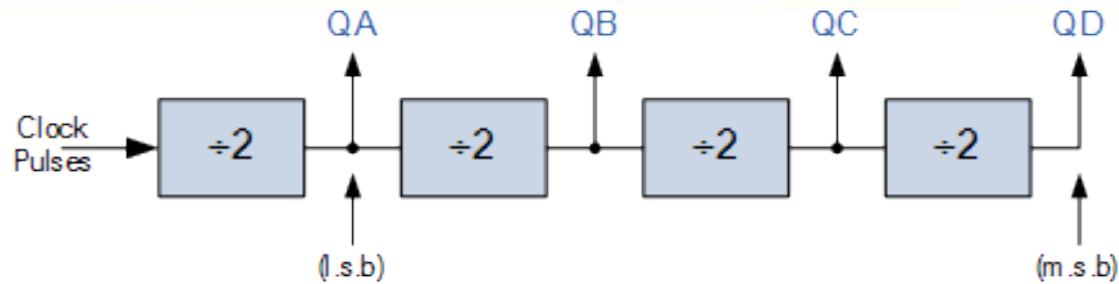


Bu devreyi proteusta çizip test edebiliriz.

Senkron Sayıcı Simulasyonu

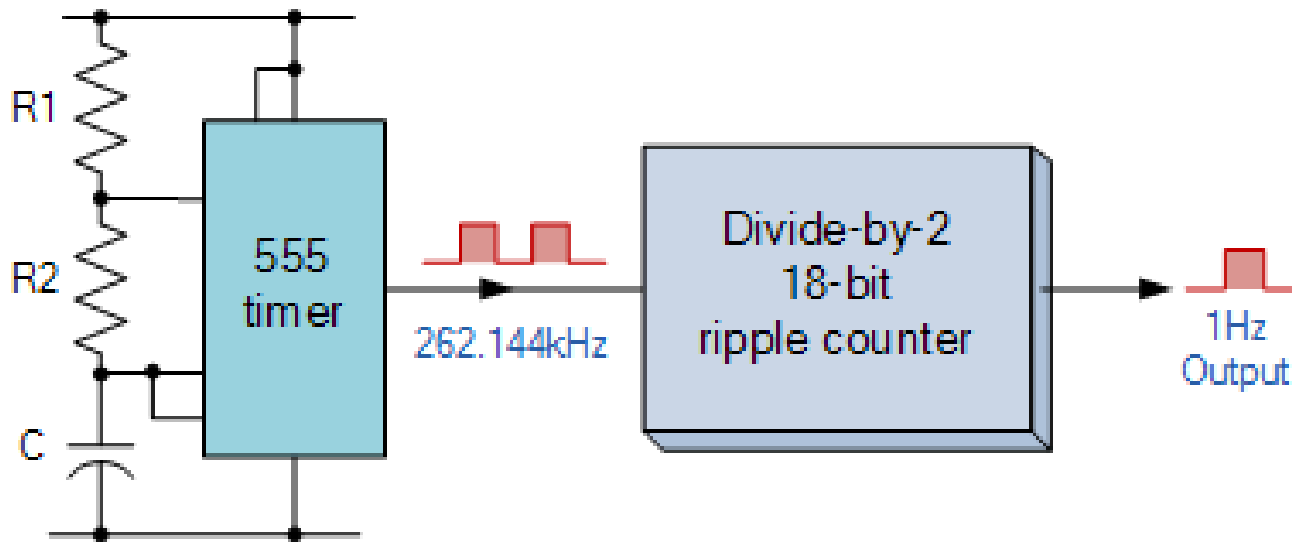


Frekans Bölme



Frekans Bölme

- 3-bit ikili sayıcı = $2^3 = 8$ (MOD-8)
- 4-bit ikili sayıcı = $2^4 = 16$ (MOD-16)
- 8-bit ikili sayıcı = $2^8 = 256$ (MOD-256)

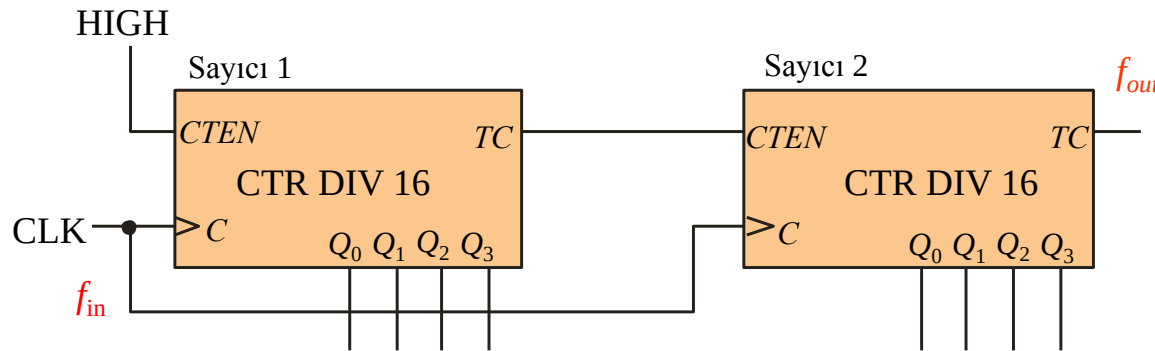


Deney 2

1. 3-artık kodda tek sayıları sayan senkron sayıcıyı J-K flip-flop kullanarak tasarlayın;
 - a. Geçiş tablosunu oluşturun,
 - b. Girişlerin fonksiyonlarını oluşturup devresini çizin,
 - c. Devreyi Proteus çizerek çalıştırın.
 - d. Devreyi deney kartınızda kurarak çalıştırın.
- (13.3.2014)

Kaskat Sayıcı Bağlantısı

Kaskat bağlantı ile daha yüksek sayma değerine sahip sayıcı elde edebiliriz. Senkron sayıcı tümdevrelerinde kaskat bağlantıda düşük değerleri sayan sayıcının terminal sayma çıkışı ile bir sonraki sayıcı izinlenerek yüksek değerli kısmın saymasına izin verilir.



Örnek:

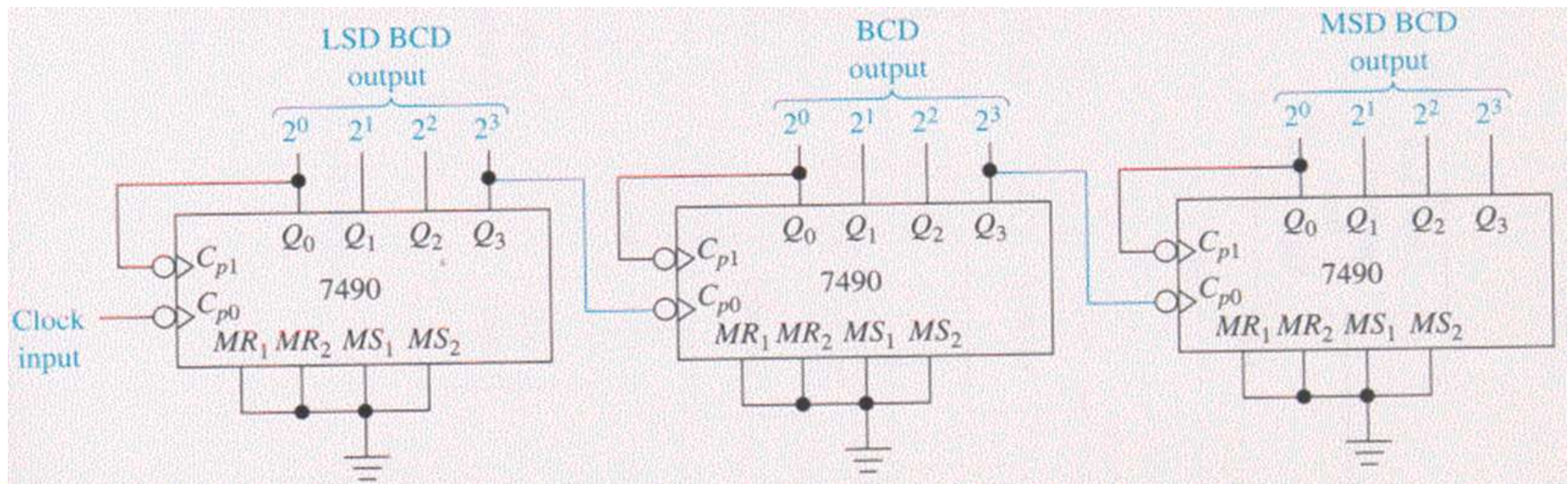
- 4 bit iki sayıcıyı kaskat bağladığımızda sayıcının mod değeri ne olur?
- Eğer $f_{in} = 100$ kHz, f_{out} ?

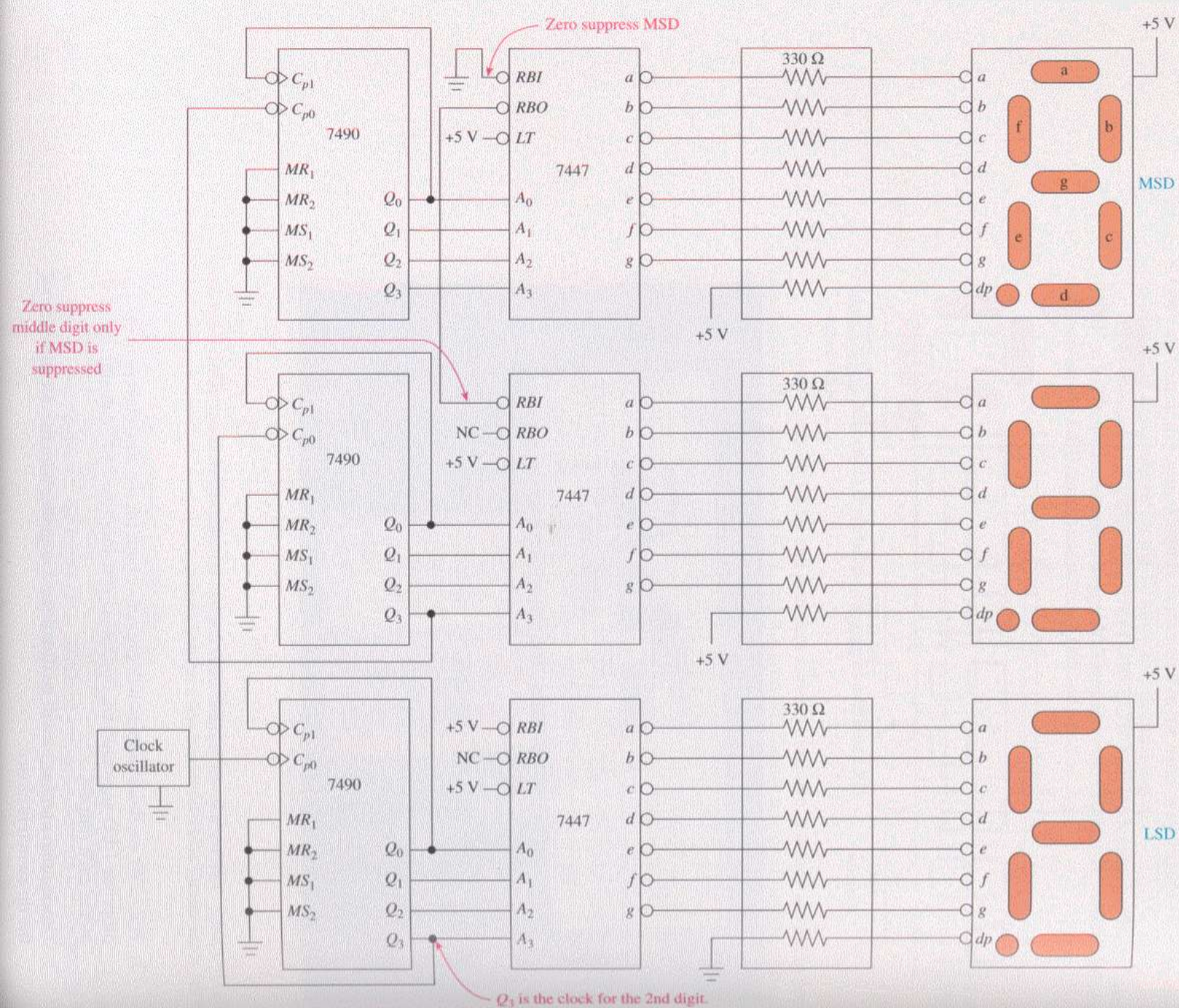
Çözüm:

- Her sayıcı giriş frekansını 16'ya böler. Sonuçta sayma modu $16^2 = 256$.
- Çıkış frekansı, $f_{ou} = 100$ kHz / 256 = 391 Hz

7490 ile 000-999 Sayıcı

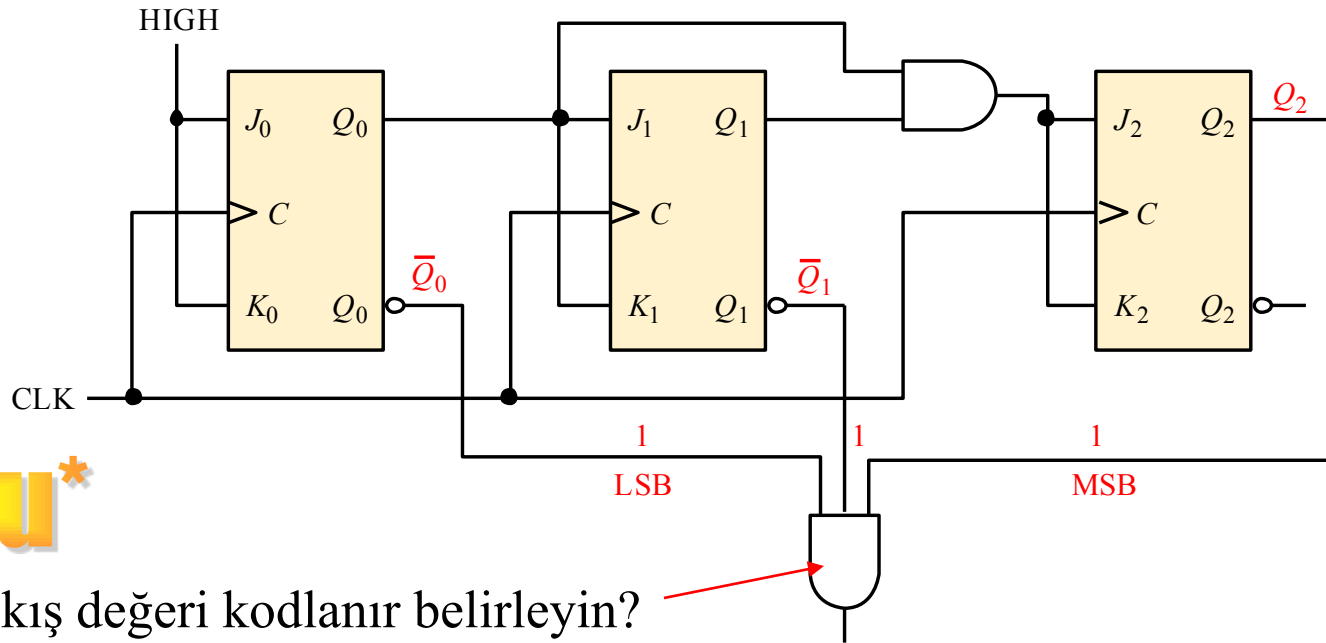
Sayıcı (1001) değerinden (0000) değerine geçişte, 2^3 veya Q_3 çıkışı Yüksekten düşüğe geçer bu düşen kenarı bir sonraki sayıcıyı tetiklemek için kullanabiliriz. Böylece her 9 saymanın sonunda bir yüksek değerli digit bir sayma yapar. Başka bir ifadeyle birler basamağının her 0...9 saymasından sonra onlar basamağı bir artar, aynı şekilde onlar basamağının her 0...9 sayması sonrası 100'ler basamağı bir artar.





Sayıc1 Kodlama Baęlantısı

Sayıcının istedięimiz sayma deęerine geldięinde bize ıkıř iřareti vermesi iin tasarladığımız baęlantıya sayıc1 kodlama olarak adlandırıyoruz. Genellikle kodlama AND geidi ile yapılır.



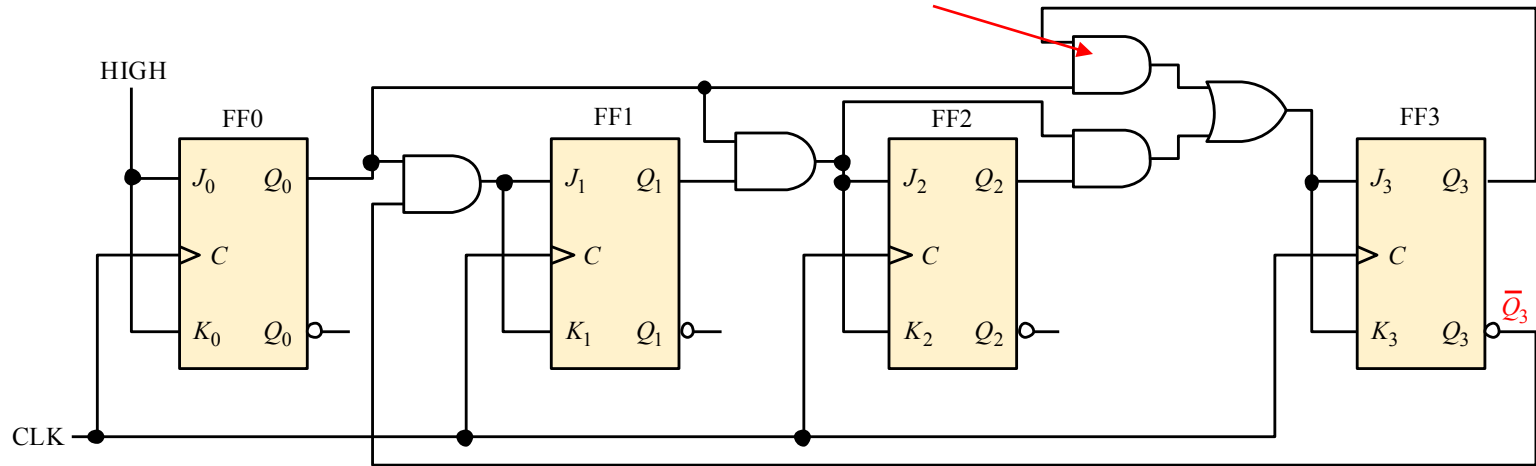
Soru*

Hangi ıkıř deęeri kodlanır belirleyin?

Parçalı Kodlama Bağlantısı

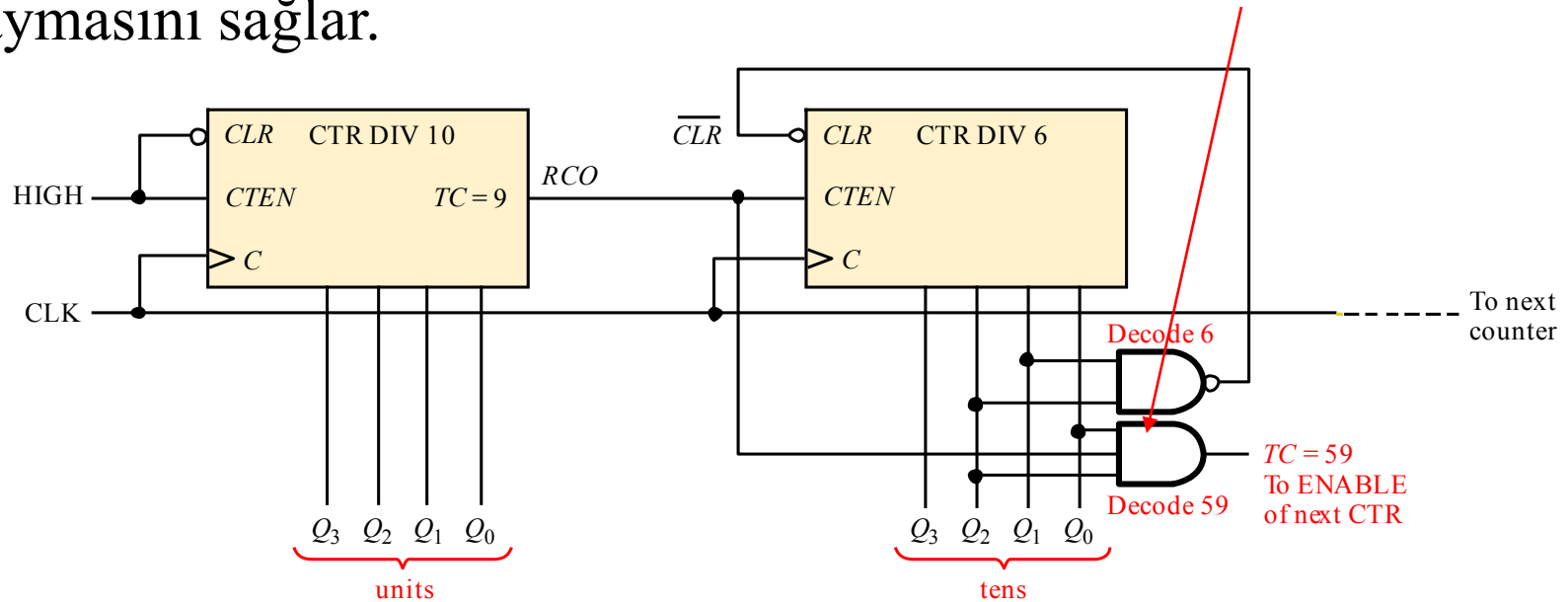
Parçalı kodlama bağlantıyı sadeleştirmek için kullanılır. Şekilde onlu sayıcının kodlaması yapılmıştır. Sayıcının sayma değeri 1001 olduğunda tekrar 0000 değerine dönmesini istiyoruz. Bu sayma değerini kodlamak için Q_3 ve Q_1 'in ilk yüksek seviye olduğu durumdur.

1001 sayma değeri iki bit denetlenerek algılanır



Sayıcı Kodlama Bağlantısı

Şekilde gösterilen Mod 60 sayıcı parçalı kodlama kullanır. Onlar basamağında 6 algılandığında sayıcıyı sıfırlar ve mod 60 saymasını sağlar.



Frekans bölünerek istediğimiz kararlı çalışan düşük frekanslı işaret elde edebiliriz. Şebeke frekansı 60 Hz olan ülkelerde osilatör yapmadan yukarıdaki devreden 1 Hz'lik kare dalga elde edilebilir.

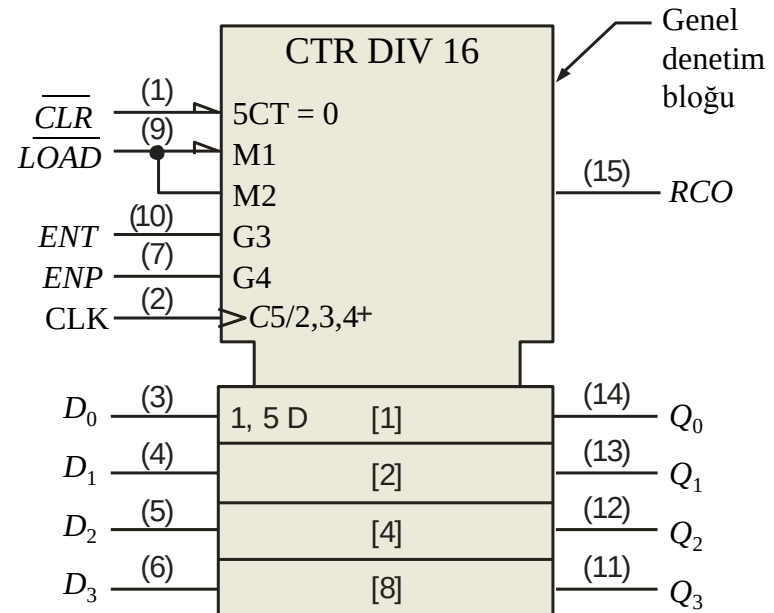
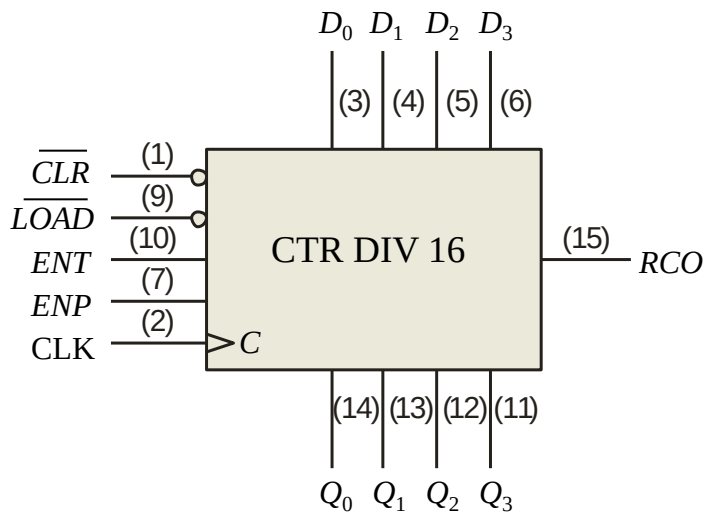
Örnek:

Çözüm:



Sayıc1 Mantık Simgeleri

Sayıc1 tümdevrelerinin aşağıda gösterilen simgeleri kullanılmaktadır.



Öğrendiğimiz Terimler

Asynchronous Eş zamanda olmayan.

Modulus Sayıcının sayma sırasında saydığı ve tekrar başa döndüğü değere verilen ad, MOD.

Synchronous Eş zamanda gerçekleşen.

Terminal count Sayıcının sayabileceği son değer

State machine Sıralı durumlar içeren mantık sistemi.

Cascade Birden fazla sayıcı tümdevresini daha büyük değer saydırmak için birinin terminal çıkışının diğerini izinleyecek şekilde bağlanmasına verilen addır.

Deney 3

1. 0-999 kaskat sayıcıyı 74hc190 kullanarak tasarlayın, Proteusta simüle edin.
2. 0-99 Yukarı/Aşağı sayıcıyı 74HC190 kullanarak tasarlayın, Proteusta çalıştırın.
3. Adım 2'deki devreyi kartınızda kurarak çalıştırın. (20.3.2014).

Proje

