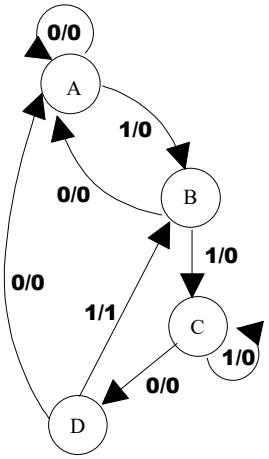
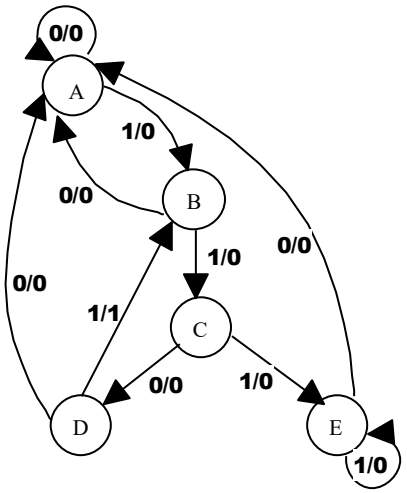
Örnek: Ard arda gelen clock saykıllarında girişin iki tane 1, sonra 0 , daha sonra da 1 olması durumunda çıkışın 1 olmasını sağlayacak Mealy türü bir sistem tasarlanması isteniyor. Çıkışın 1 olmasına neden olan 1 girişi sonraki clock saykıllarında kullanılacaktır (tekrar var).

x 0 1 1 0 1 1 0 1 0 0 1 1 1 0 1 1 0 1
z 0 0 0 0 1 0 0 1 0 0 0 0 0 0 1 0 0 1



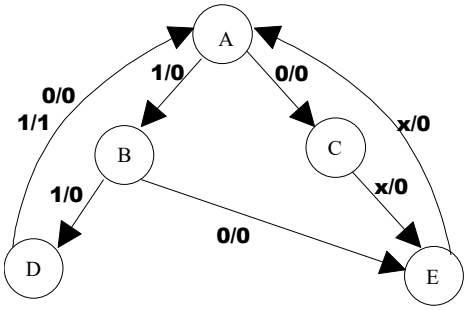
Soruya bir ilave yapalım:
Ancak ard arda girişte 3 veya daha fazla sayıda 1 gelirse bu 1 'lerin dikkate alınmaması isteniyor.

x 0 1 1 0 1 1 0 1 0 0 1 1 1 1 0 1 1 0 1
z 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1



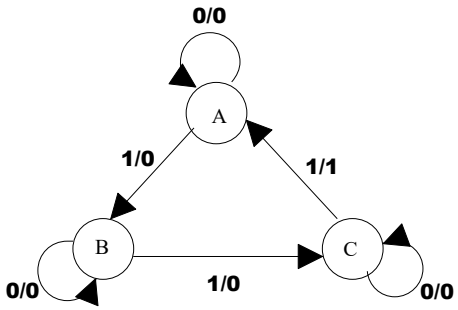
Örnek: Girişin üçlü grup olarak düşünüldüğü Mealy türü bir sistemde, grubun tüm elemanlarının 1 olması durumunda çıkışın 1 olması isteniyor.

x 1 0 1 1 1 0 1 1 1 0 1 1
z 0 0 0 0 0 0 0 0 1 0 0 0



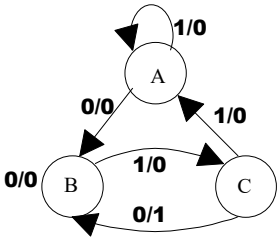
Örnek: Girişe, her üçüncü defa 1 geldiğinde, çıkışın 1 olması isteniyor. Girişteki 1'lerin ard arda olma zorunluluğu yoktur. Ayrıca, çıkışın 1 olduğu durumda, girişteki 1 değeri bir sonraki işleme katılmayacaktır (tekrar yok,Non-overlapping).

x 0 1 1 1 0 1 0 1 0 1 1 0 1 0
z 0 0 0 1 0 0 0 0 0 1 0 0 0 0



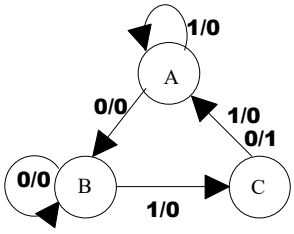
Örnek: Mealy türü bir sistemde**,** ard arda gelen clock saykıllarında giriş 010 ise çıkışın 1 olması isteniyor.

a) Girişte tekrarlama olsun.



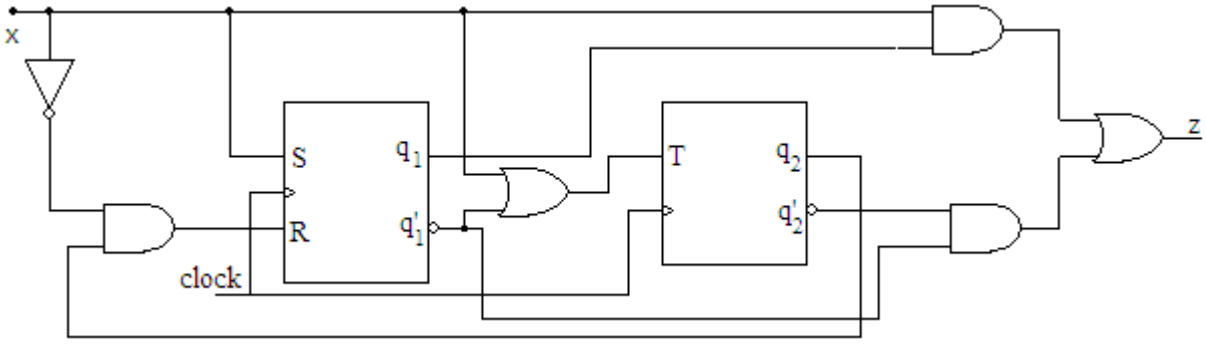
x 1 1 0 1 0 1 0 1
z 0 0 0 0 1 0 1 0

a) Girişte tekrarlama olmasın.



x 0 1 0 1 0 1 0
z 0 0 1 0 0 0 1

Örnek: Aşağıdaki devrenin durum diyagramını oluşturalım ve x'in 0 0 1 1 0 0 1 0 0 değerleri için çıkışı bulalım.



S=x R=x'.q_2 T=x+q_1' z=x.q_1+q_1'.q_2'

x=0 için S=0 ve R=q_2. Dolayısıyla q_2=0 için SR flip flobu durumunu koruyacak aksi durumda çıkış 0 olacak. x=1 için SR flip flobunun çıkışı 1 olacak. x=1 veya q_1=0 için T flip flobu mevcut durumunun tersini alacak, aksi halde durumunu muhafaza edecek. Durum tablosu;

Şimdiki Durum q_1q_2	Sonraki Durum(Q_1Q_2) x=0	x=1	Çıkış (z) x=0	x=1
00	0 1	1 1	**1**	**1**
01	0 0	1 0	**0**	**0**
10	1 0	1 1	**0**	**1**
11	**0 1**	**1 0**	**0**	**1**

(Sistemin başlangıç durumu q_1q_2=00 olarak verilmiştir.)

x 0 0 1 1 0 0 1 0 0

q_1 0 0 0 1 1 1 1 1 0 0

q_2 0 1 0 1 0 0 0 1 1 0 1

z 1 0 1 1 0 0 1 0 0 1

Örnek: Aşağıda verilen durum tablosuna göre, verilen x girişi için flip flopların durumlarını ve çıkışı belirleyiniz.

Şimdiki Durum q_1q_2	Sonraki Durum(Q_1Q_2) x=0	x=1	Çıkış (z) x=0	x=1
00	00	10	0	1
01	00	00	0	0
10	11	01	1	1
11	10	10	1	0

(Sistemin başlangıç durumu q_1q_2=00 olarak verilmiştir.)

x 0 1 0 0 1 1 1 0

q_1 0 0 1 1 1 0 0 1 1 1

q_2 0 0 0 1 0 1 0 0 1 0 1 0

z 0 1 1 1 1 0 1 1 - 1

Örnek: Aşağıda verilen durum tablosuna göre, verilen x girişi için flip flopların durumlarını ve çıkışı belirleyiniz.

Şimdiki Durum q	Sonraki Durum(Q) x=0	x=1	Çıkış (z)
A	A	B	**1**
B	D	C	**1**
C	D	C	**0**
D	**A**	**B**	**0**

x 0 1 0 1 0 1 1 1 0 1 0 0 0 0

q A A B D B D B C C D B D A A A

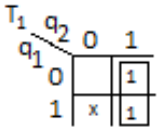
z 1 1 1 0 1 0 1 0 0 0 1 0 1 1 1 1

Sıralı olmayan sayıcı örneği ve don't care durumların incelenmesi

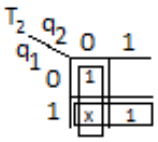
Örnek: 0 → 1 → 3

3 durum olduğu için ve 3 sayısını görüntüleyebilmek için 2 flip flop kullanmak gerekir. T tipi flip floplar kullanarak gerçekleştirimi yapalım:

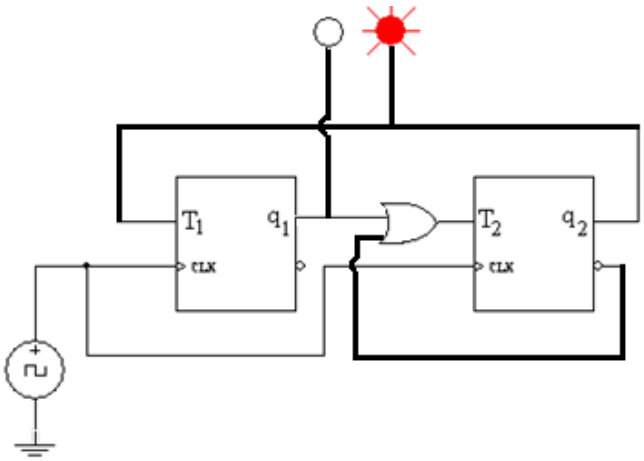
Şimdiki Durum q_1 q_2	Sonraki Durum Q_1 Q_2	T_1 T_2
0 0	0 1	0 1
0 1	1 1	1 0
1 1	0 0	1 1
1 0		x x



$T_1 = q_2$



$T_2 = q_2' + q_1$



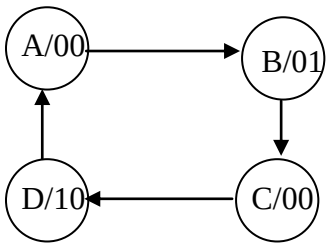
Sistem şayet don't care durum olan 10 ile başlasaydı hangi duruma gideceğini bulalım:
Karno haritası incelendiğinde T_1 için dont'care durum 0, T_2 için ise 1 kabul edilmiş. O halde T_1 flip flobu durumunu koruyacak, T_2 flip flobu ise mevcut durumunun tersini alacaktır. Yani 10 durumundan 11 durumuna gidilecektir. Daha sonra da bizim belirlediğimiz sırada sayma işlemi gerçekleşecektir. Aşağıdaki tabloda bu durum gösterilmiştir.

Şimdiki Durum q_1 q_2	Sonraki Durum Q_1 Q_2	T_1 T_2
0 0	0 1	0 1
0 1	1 1	1 0
1 1	0 0	1 1
1 0	1 1	0 1

Aşağıda değişik bir sayıcı tasarımı vardır. Genellikle sayıcıların, giriş ve z çıkışına sahip olmadan tasarlandığını söylemiştik. Ders notlarında girişe bağlı olarak ileri ya da geri sayan sayıcı örneği vardır. burada ise ekstra z çıkışları kullanılmıştır. Çünkü direkt olarak durumlardan çıkış almak bu örnekte mümkün olmamaktadır.

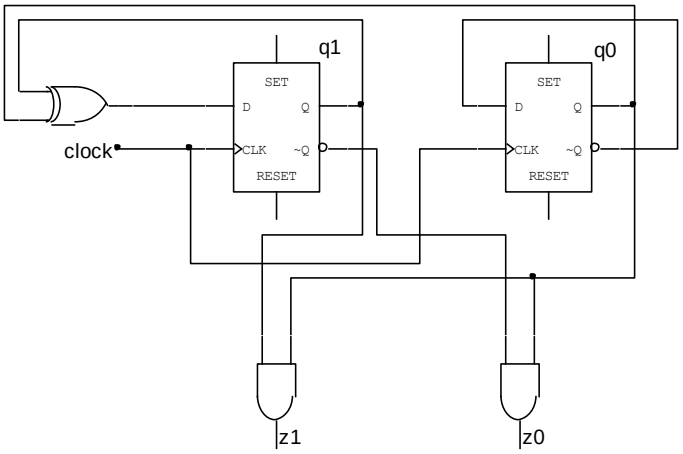
Clock darbesiyle 0-1-0-2-0-1-0-2-... şeklinde sayan bir sayıcıyı D tipi flip floplar kullanarak tasarlayınız? (İpucu: Sistem 4 duruma sahiptir. Bu sayıları göstermek için de 2 çıkışı vardır. Moore tarzı devredir.)

q1q0	Q1Q0	z1	z0
A	B	0	0
B	C	0	1
C	D	0	0
D	A	1	0



q1q0	Q1Q0	z1	z0	D1	D0
00	01	0	0	0	1
01	10	0	1	1	0
10	11	0	0	1	1
11	00	1	0	0	0

D1=q1⊕q0
D0=q0'
z1=q1.q0
z0=q1'q0



z1 ve z0 uçlarına 2 led bağlarsak önce 2 ledin de sönük olduğunu, daha sonra sağdaki ledin yandığını, daha sonra 2 ledin de sönük olduğunu ve son olarak da soldaki ledin yandığını görebiliriz. Bu işlem sürekli olarak devam edecektir. Bu tasarımda kullanılan mantıkla değişik led animasyonları (yürüyen ışık gibi) yapabiliriz.

Flip flopların başlangıç durumu nasıl ayarlanır?

Flip flopların *Preset* ve *Clear* girişleri ile ayarlama yapılır. Örnek olarak aşağıdaki flip flobun çıkışını 0 yapmak istersek Clear ucunu aktif, Preset ucunu ise pasif yapmamız gerekir. Yani Clear ucuna lojik 1, Preset ucuna da lojik 1 (değilleme işleminden dolayı pasif durum 1'dir) uygulamamız gerekir. Fakat sistemin normal çalışmasına başlayabilmesi için tekrardan Clear ucunu pasif yani 0 durumuna getirmemiz gerekecektir. Bu işlem için bir butondan faydalanılabilir.