

# EEM212 - SAYISAL DEVRE TASARIMI DERS NOTLARI

## *DERS NOTU 10: KAYDEDİCİLER*

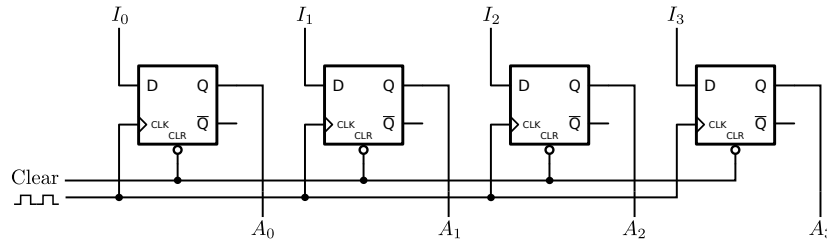
Dr. İsmail Öztürk \*

<ismail.ozturk@amasya.edu.tr>

### İçindekiler

<b>1 Giriş</b>	<b>2</b>
<b>2 Paralel Yüklemeli Kaydedici</b>	<b>3</b>
<b>3 Kaydırmalı Kaydedici</b>	<b>3</b>
3.1 SISO Kaydırmalı Kaydedici . . . . .	4
3.2 SIPO Kaydırmalı Kaydedici . . . . .	6
3.3 PISO Kaydırmalı Kaydedici . . . . .	7
3.4 PIPO Kaydırmalı Kaydedici . . . . .	9
3.5 Sola Kaydırma . . . . .	9
3.6 Çok Amaçlı Kaydırmalı Kaydedici . . . . .	11
<b>4 Kaydırmalı Kaydedici Sayıcılar</b>	<b>12</b>
4.1 Halka Sayıcı . . . . .	12
4.2 Johnson Sayıcı . . . . .	14

\* Amasya Üniversitesi Teknoloji Fakültesi EEM Bölümü  
Daha fazla bilgi için: <https://iozturk.com>



Şekil 1: 4-bit kaydedici.

## 1 Giriş

Kaydediciler (ing. *register*) dijital sistemlerdeki en temel hafıza birimidir. Burada temelden kast ettiğimiz saklanan verinin bitlerle ifade ediliyor olması ve veri yazma / okuma işleminin diğer büyük hafıza birimlerine göre çok hızlı olmasındandır. Bu hız mikroişlemci, mikrodenetleyici ya da başka dijital devrelerin doğrudan ulaşabilecekleri bir parçası olarak kullanılmalarından kaynaklıdır. Hafıza kapasitesinin küçük ve yazma / okuma hızının yüksek olmasından tahmin edebileceğiniz üzere verilerin geçici olarak hafızaya kaydedilmesi için kullanılırlar. Zaten kaydedici adı da buradan gelir.

$n$ -bit saklama kapasitesine sahip bir kaydedici  **$n$ -bit kaydedici** olarak adlandırılır.  $n$ -bitlik kaydedici elde etmek için  $n$  adet flip flop kullanılır. Kaydediciler veri saklamanın yanı sıra, veri iletimi, patern üretici, rastgele sayı üretici, sayıcı gibi pek çok uygulamada kullanılırlar. Bu nedenle, kullanım alanına göre farklı türlerde kaydediciler geliştirilmiş olup farklı türlerdeki kaydediciler  $n$  adet flip flop dışında farklı mantık kapıları (kombinasyonel devreler) içerir.

En basit kaydedici türü sadece  $n$  adet flip flopun kullanıldığı ve  $n$ -bit verinin her saat darbesiyle güncellendiği kaydedicidir. Örnek olarak bu tür 4-bit bir kaydedici Şekil 1’de görüldüğü gibidir. Şekilden görebileceğiniz üzere bu tür kaydediciler son derece basittir:  $I_i$  girişlerindeki veriler her saat darbesiyle  $A_i$  çıkışlarına aktarılır. Clear girişi ise kaydedici çıkışlarını asenkron olarak 0 yapmak için kullanılır.

Fakat, pratik uygulamalarda çoğu zaman kaydedicinin sadece 1 saat darbesi periyodu boyunca değil de farklı işlemler bitene kadar veriyi saklanması istenir. Bu durumda, Şekil 1’deki devreye “paralel yükleme” özelliği eklenmesi gerekir. Bu sayede, kaydedici her bir saat darbesi yerine bir sonraki paralel yükleme işlemine kadar flip flop çıkışlarındaki veriyi saklar. Bunu bir sonraki kısımda göreceğiz.

Bu tür basit kaydediciler dışında kaydedicilerin en ilgi çeken uygulamaları ise **kaydırmalı kaydediciler**<sup>1</sup> kullanılarak elde edilmektedir. Kaydırmalı kaydediciler flip flopların çıkışlarının bir sonraki flip flopun girişine zincirleme bir şekilde bağlanmasıyla elde edilirler. Bu sayede  $n$ -bitlik veri her saat darbesinde 1 bit kaydırılır. Kaydırmalı kaydedicileri ise üçüncü kısımda inceleyeceğiz.

<sup>1</sup>ing. *shift register*

## 2 Paralel Yüklemeli Kaydedici

Paralel yüklemeli kaydedici Şekil 1’de 4-bit için gösterilmiş olan kaydedicinin geliştirilmiş halidir. Bu tür kaydedicilerin ekstra olarak “load” adı verilen bir paralel yükleme girişleri vardır. Load girişi 0 olduğu müddetçe kaydedici çıkışındaki veriyi saklamaya devam eder. Yani sakladığı veri her saat darbesiyle güncellenmez. Load girişi 1 olduğu zaman ise  $I_i$  girişleri saat darbesiyle birlikte  $A_i$  çıkışlarına aktarılarak kaydedici güncellenmiş olur.

D FF adındaki D “data” yani veri anlamına gelmektedir. Veri saklayan kaydedicilerde D FF yaygın olarak kullanılır.

Daha önce söylemiş olduğumuz üzere dijital sistemlerde en yaygın kullanılan flip flop türü D flip floptur. Kaydedici gerçekleştirmelerinde de sıklıkla D FF kullanılır. Bu nedenle, paralel yüklemeli kaydedici gerçekleştirmesinde D FF’tan faydalanacağız. Fakat, yine hatırlayacağımız üzere D FF’ların önceki durumu koruma gibi bir özellikleri bulunmamaktadır. Dolayısıyla, D FF’ların önceki durumu koruyabilmesi için çıkışlarının bir geribesleme hattı ile girişlere bağlanması gerekir. Tabi bu geribesleme hattı sadece load 0 iken aktif olmalı; load 1 iken bu geribesleme hattı yerine kaydedici girişleri D FF girişlerine bağlanmalı.

Bahsetmiş olduğumuz bu çalışma prensibini Şekil 2’deki devre ile elde ederiz. Bu devreyi analiz etmek için sadece tek bir FF’u incelememiz yeterli çünkü diğer tüm FF bağlantıları aynı olup bu devre  $n$ -bit için genelleştirilebilir. Bu nedenle en üstteki FF bağlantılarına bakalım. Devrede load girişi 0 ise FF girişine bağlanmış olan üstteki VE kapısının çıkışı  $A_0$  olurken alttaki VE kapısının çıkışı 0 olur. Yani FF’un sonraki durum mevcut durum olan  $A_0$  olacak ve önceki durum korunacaktır. Load 1 olduğunda ise tam tersi üstteki VE kapısı çıkışı 0 olurken alttaki VE kapısının çıkışı  $I_0$  olur. Yani sonraki durum  $I_0$  girişi olur.

Diğer tüm FF’lar aynı davranacağı için büyük resme baktığımızda load 0 olduğunda kaydedici devresi önceki durumunu koruyacak; load 1 olduğunda ise kaydedici  $I_i$  girişlerini paralel olarak çıkışlara aktararak güncellenecektir. Bu devre ile kaydedici üzerinde veriyi istediğimiz süre kadar saklama imkanına sahip olmuş oluruz.

## 3 Kaydırmalı Kaydedici

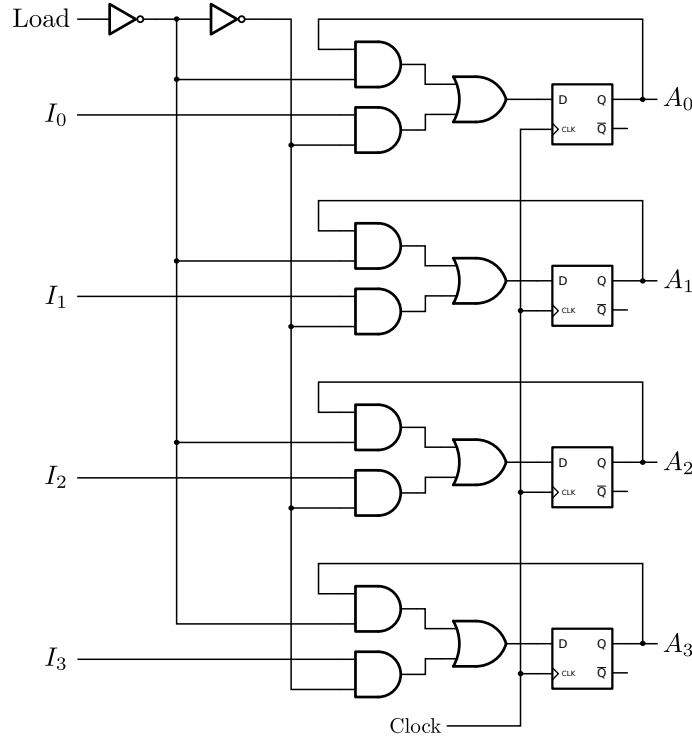
Her bir flip flop elemanında saklanan bilgiyi belirtilen yönde komşu flip flop elemanına aktarma yeteneğine sahip olan kaydedicilere kaydırmalı kaydedici adı verilir. Bu tür kaydediciler flip flopların çıkış ve girişlerinin zincir şeklinde art arda birbirlerine bağlanmasıyla elde edilir.

Tasarım tipine göre kaydırmalı kaydediciler 4 çeşittir:

1. Seri giriş / seri çıkışlı (SISO <sup>2</sup>)
2. Seri giriş / paralel çıkışlı (SIPO <sup>3</sup>)

<sup>2</sup>ing. *Serial In Serial Out*

<sup>3</sup>ing. *Serial In Parallel Out*



Şekil 2: 4-bit paralel yüklemeli kaydedici.

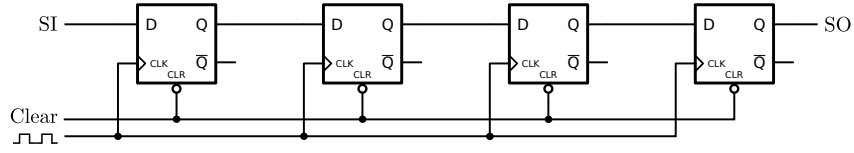
3. Paralel giriş / seri çıkışlı (PISO <sup>4</sup>)
4. Paralel giriş / paralel çıkışlı (PIPO <sup>5</sup>)

### 3.1 SISO Kaydırmalı Kaydedici

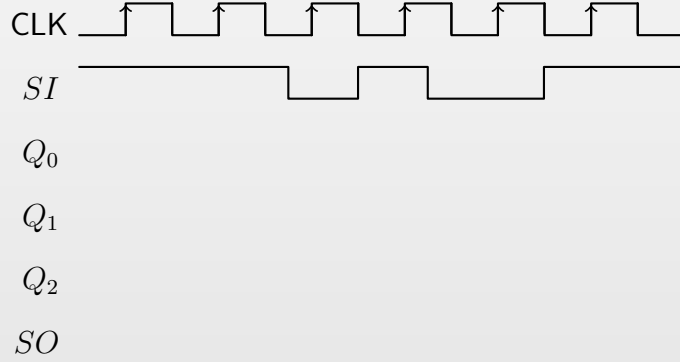
Burada örnek verilecek olan bütün kaydırmalı kaydediciler 4-bit için verilecektir. Fakat, şekillerden görebileceğiniz üzere bunlar kolaylıkla  $n$ -bit için genelleştirilebilir. Buna göre, 4-bitlik seri giriş / seri çıkışlı kaydedici Şekil 3'deki gibidir. Her bir flip flopun çıkışı bir sonraki flipflopun girişine bağlıdır. İlk flip flopun girişi seri giriş (SI); son flip flopun çıkışı ise seri çıkıştır (SO). Bu bağlantıyla kaydedicinin SI seri girişindeki bit verisi her saat darbesiyle sağdaki flip flopun çıkışına aktarılır. Yani sağa kaydırma işlemi yapılır. Şekildeki kaydedici 4-bit olduğu için SI girişi 4 saat darbesi sonrasında SO çıkışına aktarılır.

<sup>4</sup>ing. *Parallel In Serial Out*

<sup>5</sup>ing. *Parallel In Parallel Out*



Şekil 3: 4-bit SISO kaydırmalı kaydedici.

**Örnek 3.1:**

Şekil 3'deki ilk FF'un (girişi SI olan) çıkışı  $Q_0$ , ikinci FF'un çıkışı  $Q_1$ , üçüncü FF'un çıkışı  $Q_2$  olmak üzere yukarıdaki SI seri girişi için zaman diyagramının geri kalan kısmını doldurunuz (Başlangıçta tüm FF'ların clear girişi ile resetlendikten sonra clear girişinin tekrar 1 yapıldığı varsayılacaktır).

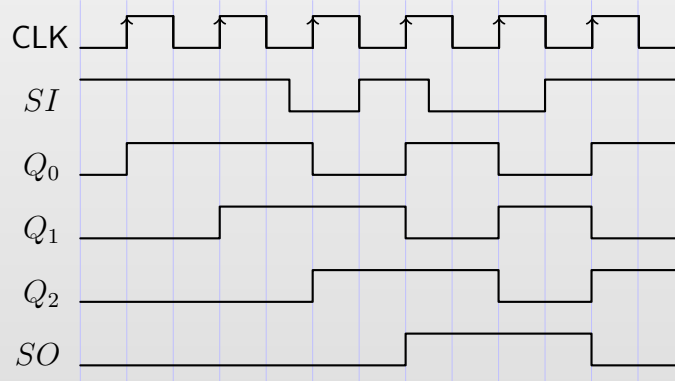
Başlangıçta FF'lar resetlendiği için saat darbesinin ilk düşük (L) kısmında tüm çıkışlar düşük yapılıır. Yani başlangıçta  $(Q_0Q_1Q_2SO)_2 = 0000$ 'dır. İlk yükselen kenarda  $SI=1$  olduğu için  $Q_0$ 'ın sonraki değeri de 1 olacaktır. Bu yükselen kenarda  $Q_0$ ,  $Q_1$  ve  $Q_2$ 'nin mevcut değerleri 0 olduğu için sırasıyla  $Q_1$ ,  $Q_2$  ve  $SO$ 'nun sonraki değerleri de 0 olacaktır. Sonuç olarak FF çıkışları yükselen kenar sonrasında 1000 olacaktır.

İkinci yükselen kenarda yine  $SI=1$  olduğundan  $Q_0$ 'ın sonraki durumu da 1 olmalıdır.  $Q_0$ 'ın mevcut durumu olan 1 ise  $Q_1$ 'in sonraki değeridir.  $Q_1$  ve  $Q_2$ 'nin mevcut durumları 0 olduğundan  $Q_2$  ve  $SO$ 'nun sonraki değerleri de 0 olmalıdır. Yani ikinci yükselen kenar sonrası FF çıkışları 1100 olur.

Üçüncü yükselen kenarda  $SI=0$ 'dır. Bu nedenle,  $Q_0$ 'ın sonraki değeri 0 olmalıdır.  $Q_0$ 'ın mevcut değeri olan 1 ise  $Q_1$ 'in sonraki değeri;  $Q_1$ 'in mevcut değeri 1 ise  $Q_2$ 'nin sonraki değeri ve son olarak  $Q_2$ 'nin mevcut değeri 0 ise  $SO$ 'nun sonraki değeri olur. Yani üçüncü yükselen kenar sonrası FF çıkışları 0110'dır.

Aynı adımları takip ettiğimizde, çıkışların dördüncü yükselen kenar sonrası 1011; beşinci yükselen kenar sonrası 0101; altıncı yükselen kenar sonrası 1010

olduğunu buluruz. Bunları, durum diyagramına çizdiğimizde aşağıdaki durum diyagramını elde ederiz:



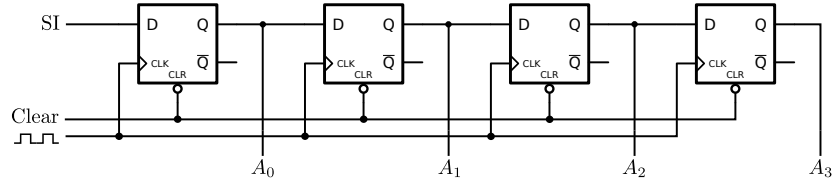
Yukarıdaki zaman diyagramından sağa kaydırma işlemi açıkça görülebilmektedir. Şimdi elde ettiğimiz sonuçları değerlendirelim: Altı adet yükselen kenar için SI değerleri sırasıyla 1, 1, 0, 1, 0, 1 şeklindedir.  $(Q_0Q_1Q_2SO)_2$  FF çıkışları ise başlangıçta 0000'dır. Her bir yükselen kenarda aslında yaptığımız bit değerlerini sağa kaydırıktan sonra en soldaki bite SI'nın o yükselen kenardaki değerini yazmaktır. Bu işlemi art arda tekrarlırsak FF çıkışlarının  $0000 \rightarrow 1000 \rightarrow 1100 \rightarrow 0110 \rightarrow 1011 \rightarrow 0101 \rightarrow 1010$  olması gerektiğini görürüz. Bu şekilde yazdığımızda da sağa kaydırma işlemi görülebilmektedir.

**Dikkat ederseniz SI girişinden verilen bir bitin SO çıkışına ulaşması için 4 adet saat darbesi tetiklemesi gerekmektedir.** Genel olarak,  $n$ -bitlik bir SISO kaydırmalı kaydedicinin giriş değerinin çıkışa ulaşması için  $n$ -adet saat darbesi gerekecektir.

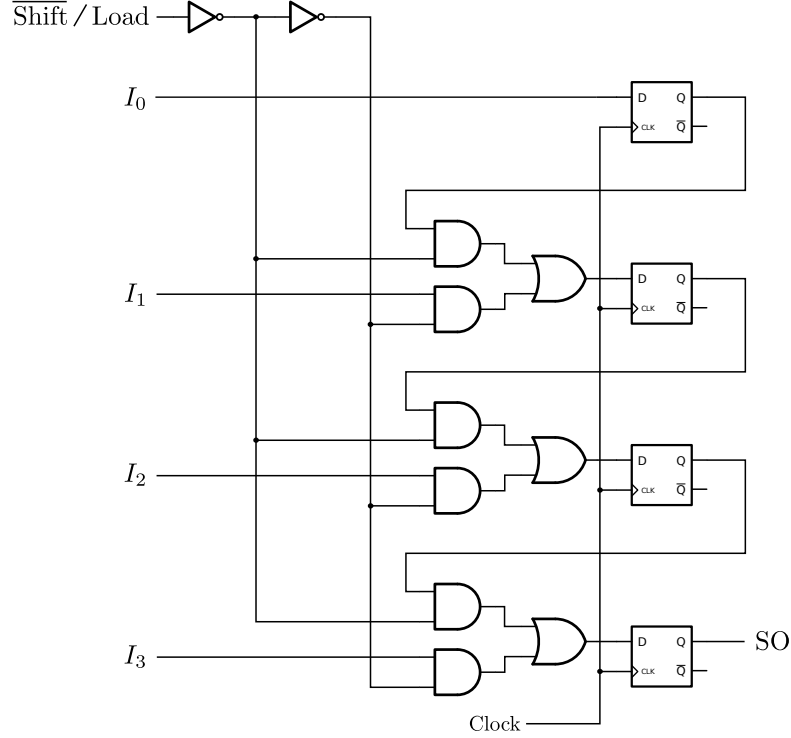
## 3.2 SIPO Kaydırmalı Kaydedici

Seri giriş / paralel çıkışlı kaydırmalı kaydedici Şekil 4'de görüldüğü gibidir. Görebileceğiniz üzere bir önceki kısımda gördüğümüz SISO kaydedicinin FF çıkışlarının her birinin ayrı paralel bir çıkış olarak kullanılmasıyla elde edilmektedir. Bu kaydedici kullanılarak seri iletilen verileri paralel hale dönüştürebiliriz. Mesela, Şekil 4'deki kaydedici seri bit iletimi yapan bir veri yolundan gelen 4-bitlik seri verileri kaydedip sonrasında bu veriyi paralel olarak başka yerlerde kullanmamıza izin verir.

Bunun dışında, çalışma prensibi SISO kaydedici ile aynıdır. Yani kaydedici tıpkı SISO gibi sağa kaydırma yapar. Örnek 3.1'de sıraladığımız 4-bitlik değerlerin aynılarını SIPO kaydedici ile elde edebiliriz. Üstüne bu değerleri çıkışlar üzerinden (led bağlayarak) gözlemleyebiliriz.



Şekil 4: 4-bit SIPO kaydırmalı kaydedici.



Şekil 5: 4-bit PISO kaydırmalı kaydedici.

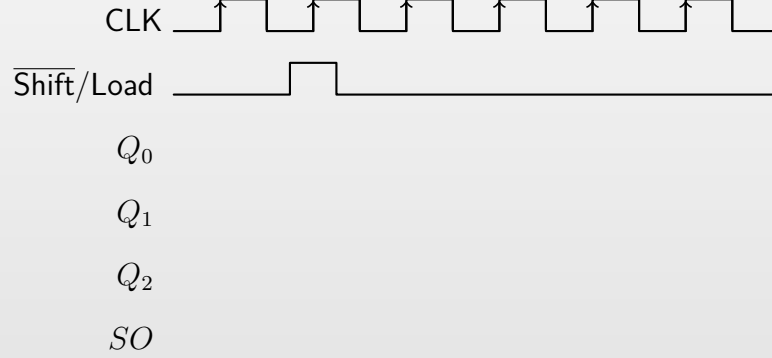
### 3.3 PISO Kaydırmalı Kaydedici

Paralel girişli / seri çıkışlı kaydırmalı kaydedici elde etmek için gerektiğinde FF girişleri üzerinden paralel yükleme yapabileceğimiz, gerektiğinde de FF'lar art arda birbirine bağlanmış gibi kaydırma işlemi yapabileceğimiz bir devre kurmamız gerekir. Bunu Şekil 2'deki paralel yüklemeli kaydedici devresinde olduğu gibi yapabiliriz. Yani bir load girişi ile paralel yükleme ve kaydırma işlemlerini kontrol edip FF girişlerine yapılması gereken farklı bağlantıları VE ile VEYA kapıları kullanarak anahtarlayabiliriz. Buna göre, PISO kaydırmalı kaydedici devresi Şekil 5'deki gibi olacaktır.

Görüldüğü üzere  $I_i$  ile ifade edilen 4 adet giriş ve bir adet SO çıkışı sırasıyla paralel girişler ve seri çıkıştır. “Shift / Load” girişi ise tek bir kontrol girişidir. Shift üzerindeki çizgiden tahmin edebileceğiniz üzere bu giriş 0 ise devre kaydırma (shift) işlemi; giriş 1 ise devre paralel yükleme (load) işlemi yapar. Bunu devreden rahatlıkla görebilirsiniz. “Shift / Load” girişi 0 olduğunda FF girişlerini bir önceki FF çıkışına

bağlayan VE kapıları aktive olur. Yani devre kaydırma yapar. Bu giriş 1 olduğunda ise  $I_i$  girişlerinin bağlı olduğu VE kapıları aktive olur ve paralel yükleme gerçekleşir. Kaydırma yapıldığında devre önceki gördüğümüz sağa kaydırma işleminin aynısını yapar. Bu devreyi kullanarak paralel bit verilerini seri hale dönüştürerek iletim yapabiliriz. Yani SIPO kaydırmalı kaydedicinin tam tersi işlev sağlar.

### Örnek 3.2:

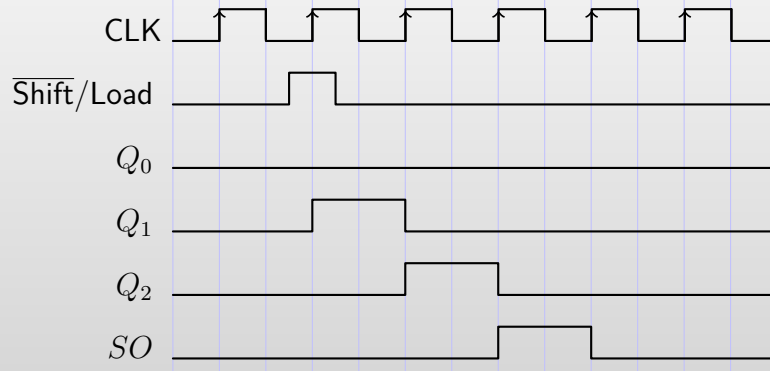


Şekil 5'deki FF çıkışları yukarıdan aşağı sırasıyla  $Q_0$ ,  $Q_1$ ,  $Q_2$  ve  $SO$  olsun. Girişler de  $(I_0I_1I_2I_3)_2 = 0100$  olsun. Buna göre, yukarıdaki zaman diyagramının geri kalan kısmını doldurunuz (Başlangıçta tüm FF'lerin clear girişi ile resetlendikten sonra clear girişinin tekrar 1 yapıldığı varsayılacaktır).

Başlangıçta  $(Q_0Q_1Q_2SO)_2 = 0000$  olduğu soruda verilmektedir. İlk yükselen kenarda  $\overline{\text{Shift/Load}}=0$  olduğu için devre sağa kaydırma yapacak ve  $I_0 = 0$  olduğu için sağa kaydırma işleminden sonra  $Q_0$ 'ın yeni değeri de 0 olacaktır. Yani FF çıkışları ilk yükselen kenardan sonra da 0000 olacaktır.

İkinci yükselen kenarda ise  $\overline{\text{Shift/Load}}=1$  olduğundan yükleme yapılacak yani FF çıkışları  $I_i$  girişlerinin değeri olan 0100 değerini alacaktır. Sonraki tüm yükselen kenarlarda  $\overline{\text{Shift/Load}}=0$  olduğundan bundan sonra hep sağa kaydırma işlemi yapılacak ve kaydırma işleminde  $Q_0$ 'ın yeni değeri hep  $I_0 = 0$  olacaktır. Buna göre, üçüncü yükselen kenardan itibaren FF çıkışları sırasıyla  $0010 \rightarrow 0001 \rightarrow 0000 \rightarrow 0000$  olmalıdır. Bu sonuçları zaman diyagramına aşağıdaki gibi çizeriz:





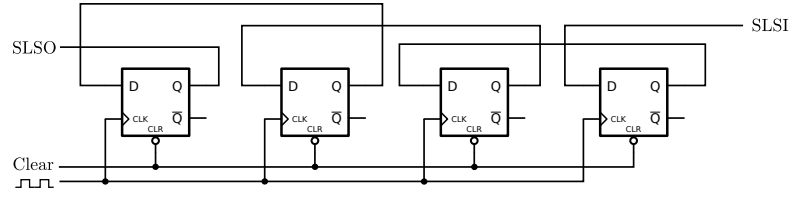
Yükleme yapıldıktan sonraki sağa kaydırma işlemi zaman diyagramından net olarak görülmektedir.

### 3.4 PIPO Kaydırmalı Kaydedici

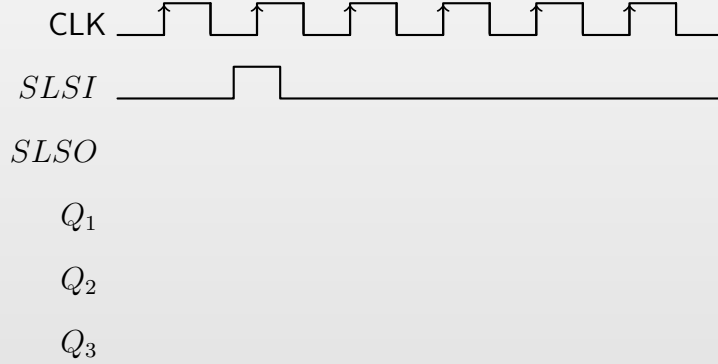
Paralel giriş / paralel çıkışlı kaydırmalı kaydedici Şekil 5'deki her bir FF çıkışının paralel olarak devre çıkışı olarak kullanılmasıyla elde edilir. Bu nedenle, PIPO kaydırmalı kaydedici devresini tekrar çizmemize gerek yoktur. Bu devrenin amacı paralel olarak bit verisini kaydettikten sonra Shift/Load girişini kullanıp veriyi kaydırmak ve en sonunda yeni veriyi paralel olarak topluca okumaktır. Örnek 3.2'teki FF çıkışlarını devre çıkışı olarak kabul edersek, bu örnek aynı zamanda PIPO kaydedici için de birebir aynı olacaktır.

### 3.5 Sola Kaydırma

Şimdiye kadar gördüğümüz tüm kaydırmalı kaydediciler sağa kaydırma (ing. *shift right*) işlemi yapıyordu. Bazen kaydedici üzerindeki verinin ters yönde de kaydırılması istenebilir. Bu durumda sola kaydırma (ing. *shift left*) işlemi nasıl yapacağımızı belirlememiz gerekir. Normalde sağa kaydırma yapmak için FF'ları sıraladığımızda bir önceki FF çıkışını bir sonraki FF girişine bağlamamız gerekir. Sola kaydırma yapmak içinse bu sıralamada bağlantıları tam tersi yapmamız gerekir. Yani bir sonraki FF çıkışını bir önceki FF girişine bağlamalıyız. Bu şekilde bir bağlantı yaptığımızda elde ettiğimiz sola kaydırmalı SISO kaydedici Şekil 6'daki gibi olacaktır. Burada SLSI (Shift Left Serial Input) seri giriş, SLSO (Shift Left Serial Output) seri çıkıştır. Diğer tür sola kaydırmalı kaydediciler de benzer şekilde elde edilecektir.



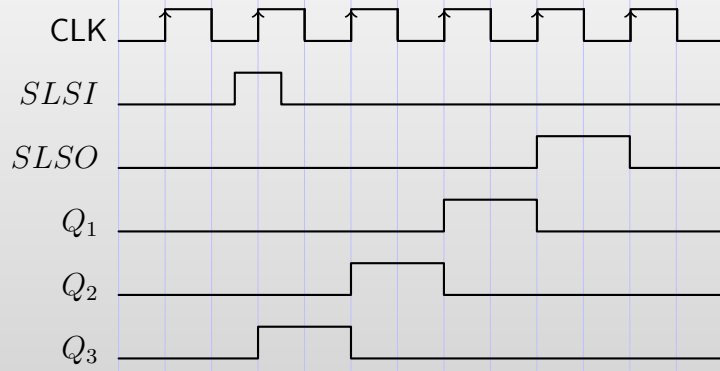
Şekil 6: 4-bit sola kaydırmalı SISO kaydedici.

**Örnek 3.3:**

Şekil 6'da SLSO çıkışlı en soldaki FF'u ilk FF kabul edip hemen sağındaki ikinci FF çıkışına  $Q_1$ , üçüncü FF çıkışına  $Q_2$ , dördüncü FF çıkışına  $Q_3$  diyelim. Buna göre yukarıdaki SLSI seri girişi için zaman diyagramının geri kalan kısmına doldurunuz (Başlangıçta tüm FF'ların clear girişi ile resetlendikten sonra clear girişinin tekrar 1 yapıldığı varsayılacaktır).

Sola kaydırmada SLSO,  $Q_1$  ve  $Q_2$ 'nin sonraki değerleri sırasıyla  $Q_1$ ,  $Q_2$  ve  $Q_3$ 'ün mevcut değerleri olacaktır.  $Q_3$ 'ün sonraki değeri ise SLSI girişi tarafından belirlenecektir. Buna göre, başlangıçta 0000 olan  $(SLSOQ_1Q_2Q_3)_2$  değerleri her bir yükselen kenarda şu şekilde güncellenecektir: 0000  $\rightarrow$  0000  $\rightarrow$  0001  $\rightarrow$  0010  $\rightarrow$  0100  $\rightarrow$  1000  $\rightarrow$  0000. Buradan sola kaydırma işleminin nasıl daha önce görmüş olduğumuz sağa kaydırma işleminin tam tersi olduğunu görebilirsiniz.

Bu değişimi zaman diyagramına işlediğimizde ise aşağıdaki gibi bir zaman diyagramını elde ederiz:



Zaman diyagramından görülebileceği üzere, giriş verisi daha önce görmüş olduğumuz zaman diyagramlarının aksine en soldaki FF'a doğru kaydırılmaktadır.

### 3.6 Çok Amaçlı Kaydırmalı Kaydedici

Çok amaçlı kaydırmalı kaydedici (ing. *universal shift register*) yukarıda bahsettiğimiz sağa kaydırma, sola kaydırma, paralel yükleme, paralel çıkış gibi işlevlerin hepsini bir arada yapabilen kaydedicidir. FF girişlerine  $4 \times 1$  mux bağlayarak elde edilir. Mux kontrol bitlerine göre aşağıdaki gibi çalışması istenir:

$S_1$	$S_0$	Çalışma Modu
0	0	Önceki Durumu Koru
0	1	Sağa Kaydır
1	0	Sola Kaydır
1	1	Paralel Yükleme Yap

Önceki durumu korumak için FF girişine aynı FF'un çıkışını bağlamak gerekir. Yani çok amaçlı kaydırmalı kaydedicinin  $S_1 = 0$ ,  $S_0 = 0$  için önceki durumu koruyabilmesi adına  $I_0$  mux girişine aynı FF'un çıkışı bağlanmalıdır.

Sağa kaydırma yapmak için (daha önce görmüş olduğumuz üzere) FF girişine bir önceki FF'un çıkışı bağlanmalıdır. Yani çok amaçlı kaydırmalı kaydedicinin  $S_1 = 0$ ,  $S_0 = 1$  için sağa kaydırma yapabilmesi adına  $I_1$  mux girişine bir önceki FF'un çıkışı bağlanmalıdır.

Sola kaydırma içinse (daha önce görmüş olduğumuz üzere) FF girişine kendinden bir sonraki FF'un çıkışı bağlanmalıdır. Yani çok amaçlı kaydırmalı kaydedicinin  $S_1 = 1$ ,  $S_0 = 0$  için sola kaydırma yapabilmesi adına  $I_2$  mux girişine bir sonraki FF'un çıkışı bağlanmalıdır.

Paralel yükleme yapmak içinse FF girişlerine paralel girişler bağlanmalıdır. Yani çok amaçlı kaydırmalı kaydedicinin  $S_1 = 1$ ,  $S_0 = 1$  için paralel yükleme yapmasını sağlamak adına  $I_3$  mux girişine paralel giriş bağlanmalıdır.

Bahsedilen bu bağlantıları yaptığımızda çok amaçlı kaydırmalı kaydediciyi Şekil 7'deki gibi elde ederiz. Çok amaçlı kaydırmalı kaydedicide bitler  $A_0A_1A_2A_3$  sırasıyla saklanmaktadır. Sola kaydırmada veri  $A_0$  yönüne doğru kaydırılır. Bu durumda  $A_3$ 'ün yeni değeri SLSI (Shift Left Serial Input) girişi tarafından belirlenir. Sağa kaydırmada ise işlem tam tersidir. Kaydırma  $A_3$  yönüne doğrudur ve  $A_0$ 'ın yeni değeri SRSI (Shift Right Serial Input) girişi tarafından belirlenir. Paralel yüklemde ise  $I_i$  girişleri paralel  $A_i$  çıkışlarına aktarılır. Önceki durumu koruma modunda ise  $A_i$  çıkışları değiştirilmez. İşte çok amaçlı kaydedici bu şekilde çalışır.

## 4 Kaydırmalı Kaydedici Sayıcılar

Kayırmalı kaydediciler pratikte sıklıkla sayıcı olarak kullanılırlar. Bu kısımda kaydırmalı kaydedicilerin nasıl sayıcı olarak kullanıldıkları incelenecektir.

### 4.1 Halka Sayıcı

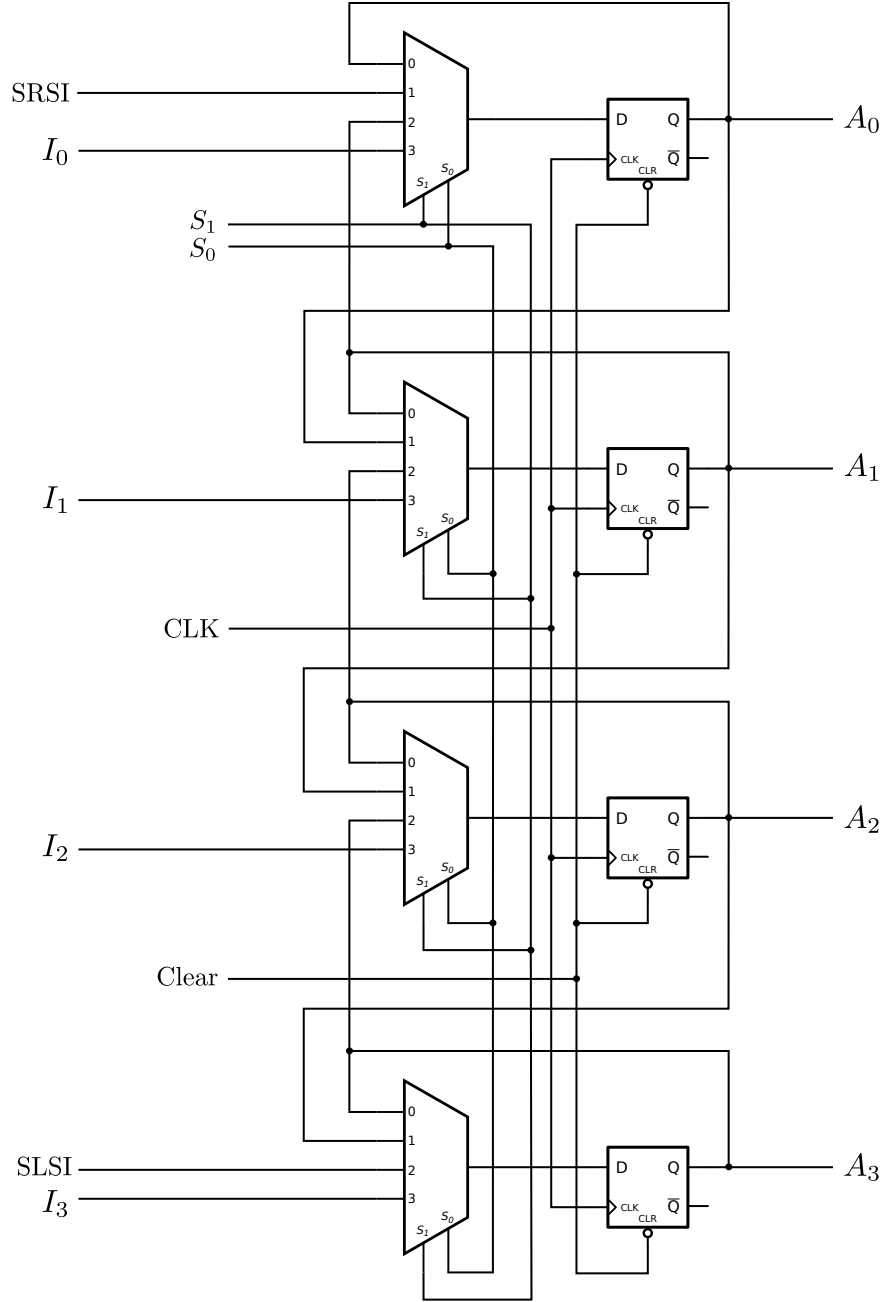
Halka sayıcı  $n$ -bitlik bir kaydırmalı kaydedicinin son FF çıkışının ilk FF girişine bağlanmasıyla elde edilir. Bu sayede kaydırma işlemi halka şeklinde olacaktır. Yani en sondaki bit kaydırıldıktan sonra tekrar başa dönecektir. Halka sayıcıda sayma işlemi aynı anda sadece tek bir FF çıkışının 1 olmasıyla yapılır. Bunu sağlamak içinse FF'ların başlangıç durumları uygun bir biçimde ayarlanmalıdır. FF'ların başlangıç durumlarını ayarlamak için daha önce preset ve clear asenkron girişlerini kullandığımızdan bahsetmiştik. Şimdi bu girişleri pratikte nasıl kontrol edeceğimizi göreceğiz.

Örnek olarak Şekil 8'deki 4-bitlik halka sayıcı devresini inceleyelim. Görebileceğiniz üzere RC devresinin çıkışı üç adet clear girişine ve bir adet preset girişine bağlanmıştır. Devreye ilk enerji verildiğinde kapasitör üzerinde  $0V$  görüleceği için üç FF çıkışı resetlenecek (çıkışları 0 yapılacaktır); en baştaki FF ise set edilecektir (çıkışı 1 yapılacaktır). Fakat, çok kısa süre içerisinde kapasitör şarj olacağı için clear ve preset girişlerine  $+V_{CC}$  (yani lojik 1) verileceğinden FF'lar normal çalışmasına devam edecektir. İşte FF'ların başlangıç durumları RC devresi kullanılarak bu şekilde ayarlanır.

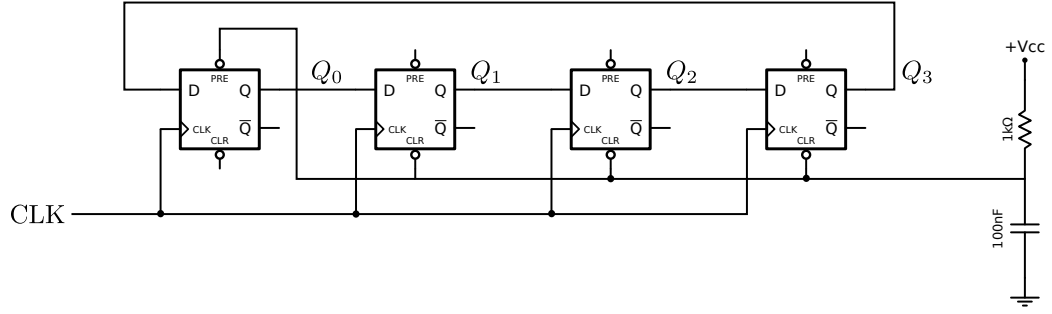
Şekil 8'ye göre başlangıçta  $(Q_0Q_1Q_2Q_3)_2$  değeri 1000 olacaktır. Bağlantı şekli kaydırmalı kaydedici olduğu için her yükselen kenarda sağa doğru kaydırma olacaktır. Fakat kaydırma yaparken geribesleme nedeniyle en sondaki FF değeri tekrar başa döndürülecektir. Buna göre, FF çıkışları şu şekilde değişecektir:

$$1000 \rightarrow 0100 \rightarrow 0010 \rightarrow 0001 \rightarrow 1000 \rightarrow 0100 \rightarrow \dots$$

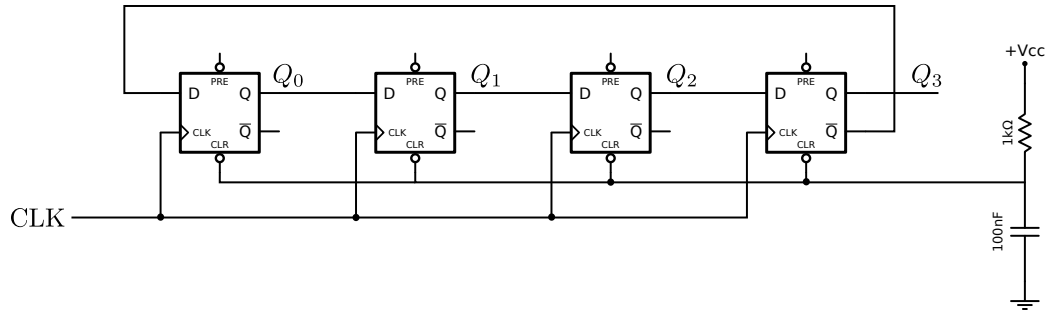
Görebileceğiniz üzere her 4 saat darbesinde halka sayıcı tekrar başa dönmektedir. Yani Şekil 8'deki sayıcının 4 farklı durumu vardır. **Genel olarak,  $n$ -bitlik bir halka sayıcının  $n$  farklı durumu vardır.**



Şekil 7: Çok amaçlı kaydırmalı kaydedici (4-bit).



Şekil 8: 4-bit halka sayıcı.



Şekil 9: 4-bit Johnson sayıcı.

## 4.2 Johnson Sayıcı

Johnson sayıcı bir önceki kısımda görmüş olduğumuz halka sayıcının Şekil 9'deki gibi modifiye edilmiş halidir. Görebileceğiniz üzere iki değişiklik yapılmıştır. Öncelikle ilk FF'un preset girişine bağlı olan RC devresi, o FF'un clear girişine bağlanmıştır. Bu artık tüm FF'ların başlangıçta resetlenerek 0 yapılacağı anlamına gelmektedir. Diğer değişiklik olarak ise artık geribesleme hattı  $Q_3$  çıkışı yerine  $\overline{Q_3}$  çıkışından alınmaktadır. Bu ise her sağa kaydırma işleminde  $Q_0$ 'ın yeni değerinin  $Q_3$  değerinin tersi olacağı anlamına gelmektedir.

Elde ettiğimiz bu iki sonucu birleştirdiğimizde,  $(Q_0Q_1Q_2Q_3)_2$  değerinin her yükselen kenarla birlikte aşağıdaki gibi değişeceğini anlarız:

$$0000 \rightarrow 1000 \rightarrow 1100 \rightarrow 1110 \rightarrow 1111 \rightarrow 0111 \rightarrow 0011 \rightarrow 0001 \rightarrow 0000 \rightarrow \dots$$

Görebileceğiniz üzere Şekil 9'deki Johnson sayıcı 8 saat darbesinden sonra başa dönerek kendisini tekrar etmektedir. Yani 8 farklı duruma sahiptir. Genel olarak,  $n$ -bitlik bir Johnson sayıcının  $2n$  farklı durumu vardır.