EEM212 - SAYISAL DEVRE TASARIMI DERS NOTLARI

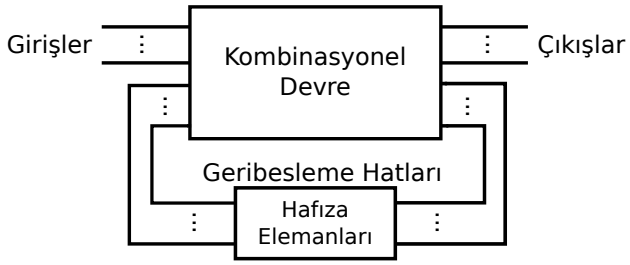
DERS NOTU 5: ARDIŞIL DEVRELER I: HAFIZA ELEMANLARI

Dr. İsmail Öztürk *

<ismail.ozturk@amasya.edu.tr>

İçindekiler

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



Şekil 1: Ardışıl devrelerin blok diyagramı.

1 Ardışıl Devrelere Giriş

Daha önce dijital devrelerin kombinasyonel ve ardışıl olmak üzere ikiye ayrıldığından bahsetmiştik. Kombinasyonel devreler çıkışları sadece girişlerin o anki değerlerinin kombinasyonlarına bağlı olan dijital devrelerdi. Ardışıl devrelerde ise çıkışlar hem girişlerin mevcut değerlerine, hem de önceki değerlerine bağlı olarak belirlenir. Girişlerin önceki değerlerini hesaplamada kullanabilmek için bu değerleri hafızada tutan hafıza elemanlarından faydalanılır. Böylelikle, mevcut ve önceki değerleri kullanarak sonraki çıkış değerleri hesaplanır. Bu hesaplama yapılırken yine bildiğimiz kombinasyonel devrelerden faydalanılır. Buna göre, ardışıl bir devrenin blok diyagramı Şekil 1 'deki gibi olacaktır. Girişlerin önceki değerlerini kombinasyonel devre işlemine sokmak için kullandığımız hafıza elemanı bağlantıları, şekilde görülen geribesleme hatlarını oluşturur.

Ardışıl devrelerde hafıza elemanının aldığı değere **durum** adı verilir. Ya girişlerde yapılacak değişikliklerle ya da harici bir sinyal uygulanarak bu hafıza elemanlarının güncellenmesi, mevcut durumdan sonraki duruma geçişi sağlar. Hafıza elemanının sonraki durumu harici girişlerin yanı sıra hafıza elemanının mevcut durumuna bağlıdır. Bu şekilde mevcut durumlara bağlı olarak sonraki durumların hesaplanması bize ardışıl bir ilişki verir. Mesela, hafıza elemanının **başlangıç durumu**nu Q_0, sonraki durumunu Q_1 ile etiketlersek Q_1 değerini Q_0 değerini (ve diğer değişkenleri) kullanarak hesaplarız. Bir sonraki durum olan Q_2 ise Q_1 değeri kullanılarak hesaplanır. Hesaplamalara devam ettikçe, Q_0, Q_1, Q_2, Q_3, ... şeklinde giden ardışıl ilişkiye sahip bir dizi elde ederiz. Genel olarak, mevcut durum Q_n ile ifade edilirken, hafıza elemanının güncellenmesiyle elde edilecek sonraki durum Q_{n+1} ile ifade edilir. Mevcut ve sonraki durumu ifade etmek için sırasıyla $Q(n)$ ve $Q(n+1)$ notasyonları da benzer şekilde kullanılmaktadır.

Ardışıl devreler

1. Senkron ardışıl devreler
2. Asenkron ardışıl devreler

olmak üzere ikiye ayrılır.

Senkron ardışıl devrelerde hafıza elemanlarının güncellenmesi için harici periyodik bir sinyal uygulanır. Periyodik ve kare dalga olmasından dolayı bu sinyallere **saat darbeleri** adı verilir. Senkron devrede bulunan tüm hafıza elemanları bu saat darbelerine göre aynı anda (senkronize bir şekilde) sonraki duruma geçerler.

Asenkron ardışıl devrelerde ise hafıza elemanlarının güncellenmesi için harici periyodik bir sinyal kullanılmaz. Girişlerde yapılacak bir değişiklik otomatik olarak hafıza elemanının güncellenmesine neden olur. Asenkron devreler mantık kapılarında bulunan yayılma (propagasyon) gecikmelerini kullanan zaman-gecikmeli devrelerdir.

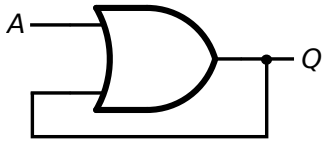
Tanım 1.1: Yayılma (Propagasyon) Gecikmesi

Mantık kapılarının girişinde yapılan bir değişikliğin çıkışa yansıması için belli bir süre gerekir. İşte bu süreye yayılma (propagasyon) gecikmesi adı verilir. Bir dijital devrede giriş ve çıkış arasındaki hatta pek çok mantık kapısı kullanılır. Dolayısıyla, devrenin girişinde yapılan bir değişikliğin devrenin çıkışına yansıması için geçecek süreye de toplam yayılma gecikmesi adı verilir. Yayılma gecikmesi ardışıl devre tasarımında dikkat edilmesi gereken en önemli kriterlerdendir.

Senkron ve asenkron devrelerin karşılaştırması aşağıdaki gibidir:

Asenkron Devreler	Senkron Devreler
Hafıza elemanları farklı farklı zamanlarda güncellendiği için tasarımları ve analizleri zordur.	Hafıza elemanları aynı anda güncellendiği için tasarımları ve analizleri kolaydır.
Güncellemeler anlık ve bireysel yayılma gecikmelerine bağlı olduğu için hızlıdırlar.	Güncellemeler ortaktır ve devrede bulunan en yavaş yayılma gecikmesinden daha uzun sürede yapılmalıdırlar. Bu nedenle yavaştırlar.
Birden fazla girişin aynı anda değişmesi durumunda devrenin nasıl çalışacağı belirsizdir. Bu duruma "**race condition**" adı verilir.	Hafıza elemanları giriş değerleri ile güncellenmediğinden race condition asenkron devrelerdeki kadar önemli bir sorun değildir.
Güç tüketimi düşüktür.	Tüm hafıza elemanlarına dağıtılması gereken periyodik bir sinyalin varlığı güç tüketimini arttırır.

Günümüzde asenkron devrelerin kullanımı oldukça sınırlıdır. Ardışıl dijital devrelerin yalnızca yüksek hız gerektiren kısımlarında kullanılırlar. Onun dışında tasarım ve analiz kolaylığı, race condition probleminden fazla etkilenmemeleri gibi nedenler yüzünden diğer tüm ardışıl devrelerde senkron devre yapısı tercih edilir. Fakat, senkron ardışıl devreler oluşturmak için öncelikle asenkron devreler oluşturmak gerekir. Bu nedenle, senkron devreleri incelemeye başlamadan önce asenkron devreler hakkında bilgi sahibi olmamız gerekir.



Şekil 2: Basit bir asenkron ardışıl devre örneği.

Basit bir asenkron ardışıl devre örneği Şekil 2 'deki gibidir. Şekilden görebileceğiniz üzere devre tek giriş ve tek çıkışlıdır. Çıkış VEYA kapısının bir girişine bağlı olduğu için devrenin Şekil 1 'deki gibi bir geribesleme hattı vardır. Peki bu geribesleme hattında kullanılan hafıza elemanı nerede diye merak edebilirsiniz. Bu geribesleme hattındaki hafıza elemanı aslında VEYA kapısının kendisidir. VEYA kapısının yayılma gecikmesi, girişte yapılacak bir değişikliğin çıkışa yansımasına kadar geçen sürede Q değerinin mevcut durumunun saklanmasına; yani kapının bir nevi hafıza elemanı gibi davranmasına neden olur. Hafıza elemanının güncellenmesi A girişinde yapılacak değişiklikler ve yayılma gecikmesine bağlı olduğu için devrenin asenkron olduğunu anlarız.

Bu devrenin analizini yapabilmek için önce Q çıkışının mevcut durumunu Q_n olarak etiketleriz. Devreye baktığımızda Q_n ve A girişinin VEYA işlemine sokularak Q çıkışının sonraki değeri olan Q_{n+1}'in hesaplandığını görürüz. Bunu aşağıdaki eşitlikle ifade ederiz:

$$Q_{n+1} = A + Q_n \tag{1}$$

Bu tür eşitlikler genel olarak **fark denklemi** olarak adlandırılır. Eşitlikten görülebileceği üzere çıkışın ne olacağı A girişinin değeri ve Q çıkışının mevcut durumuna bağlıdır. Şimdi A ve Q_n değerlerinin alabileceği farklı değerler için devrenin sonraki değerlerinin ne olacağını inceleyelim. $A = 0$ ve $Q_0 = 0$ olduğunda Eşitlik 1'i kullandığımızda sonraki değer $Q_1 = 0$ olacaktır. Yani bu durumda Q çıkışının sonraki durumu da 0 olmaktadır. Sonraki değerleri $\rightarrow$ ile ifade edersek sonraki durumları yine Eşitlik 1 'i kullanarak aşağıdaki gibi elde ederiz:

$$\text{(i)} \quad A = 0 \text{ için} \quad Q_0 = 0 \rightarrow Q_1 = 0 \rightarrow Q_2 = 0 \rightarrow \ldots$$

Gördüğünüz üzere $A = 0$ olduğu müddetçe Q'nun sonraki değerleri de sürekli 0 olmaktadır. Çıkışların bu şekilde sabit kaldığı durumlara **kararlı durum** adı verilir. $A = 0$ için $Q_0 = 1$ olursa

$$\text{(ii)} \quad A = 0 \text{ için} \quad Q_0 = 1 \rightarrow Q_1 = 1 \rightarrow Q_2 = 1 \rightarrow \ldots$$

bu sefer Q çıkışı 1 değerinde kararlı hale gelir. $A = 1$ için başlangıç durumu $Q_0 = 0$ olursa

$$\text{(iii)} \quad A = 1 \text{ için} \quad Q_0 = 0 \rightarrow Q_1 = 1 \rightarrow Q_2 = 1 \rightarrow \dots$$

elde edilir. Çıkış yine 1 değerinde kararlı hale gelmiştir. Son olarak $A = 1$, $Q_0 = 1$ durumunu incelersek yine

$$\text{(iv)} \quad A = 1 \text{ için} \quad Q_0 = 1 \rightarrow Q_1 = 1 \rightarrow Q_2 = 1 \rightarrow \dots$$

elde ederiz.

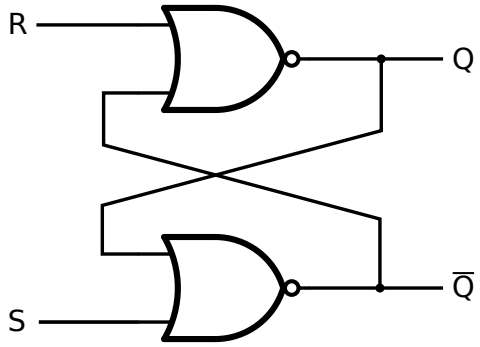
Bu sonuçlara göre, Şekil 2 'deki devrenin çıkışının sadece ve sadece $A = 0$ ve $Q_n = 0$ olursa 0 olacağını diğer tüm durumlarda çıkışın 1 değerinde sabit kalacağını görürüz. Buna göre bu devreyi kurduğunuzda ilk çalıştırdığınızda $A = 0$ yaparsanız $Q_n = 0$ olacağından devrenin çıkışı sürekli 0 olur. $A = 0$ olduğu müddetçe bu durumda kalınır. Bu durumdayken $A = 1$ yaparsanız yukarıdaki (iii) numaralı durumu elde ederiz ki bu durumda Q çıkışı 1 olur ve $A = 1$ olduğu müddetçe bu durum korunur. Eğer bu durumdayken A tekrar 0 yapılırsa (ii) numaralı duruma geçeriz ki Q yine 1 değerini korur. Tekrar $A = 1$ yaparsak, (iv) numaralı duruma geçeriz fakat $Q = 1$ yine değişmez. Gördüğünüz üzere, tekrar $Q = 0$ yapabilmemiz mümkün değildir; A'nın değerinden bağımsız olarak $Q = 1$ değeri sürekli korunacaktır.

Yukarıda yapmış olduğumuz analize göre, Şekil 2 'deki devreyi kurduğunuzda ilk çalıştırmada $A = 0$ verirseniz $Q_0 = 0$ olacağı için çıkışınız $Q = 0$ olacaktır. Eğer, kısa süreli bile olsa bir kez $A = 1$ yaptığınızda çıkış sürekli $Q = 1$ olacaktır. Tekrar $A = 0$ yapsanız bile çıkış $Q = 0$ olmayacaktır.
Bu ardışıl devrelerin kombinasyonel devrelerden ne kadar farklı çalıştığını bize gösterir. Kombinasyonel devrelerde bir çıkışı aynı giriş değerleriyle elde edebiliriz. Fakat, ardışıl devrelerde önceki değerlere bağlılık söz konusu olduğundan aynı girişler bu örnekteki $A = 0$'da olduğu gibi farklı çıkış değerleri verebilir.

Şimdi ardışıl devrelerde kullanılan hafıza elemanlarının neler olduğunu göreceğiz.

2 Hafıza Elemanları

Ardışıl devrelerde kullanılan hafıza elemanları da yapılarına göre senkron ve asenkron olarak ikiye ayrılır. Asenkron hafıza elemanları **tutucu** veya **latch** olarak adlandırılırken, senkron hafıza elemanları **flip flop** olarak adlandırılır. Senkron ardışıl devreler oluşturmak için kullanılan flip flopların iç yapısı tutuculara bağlı olduğu için öncelikle aslında asenkron ardışıl devreler olan tutucuların çalışma prensibini inceleyeceğiz. Sonrasında ise tutucular kullanarak flip flopları elde edeceğiz.



Şekil 3: VEYADEĞİL tabanlı SR Latch devresi.

2.1 Latch'ler (Tutucular)

Latch'ler kontrol girişlerine karşılık çıkışta Q ve $\overline{Q}$ ile ifade edilen iki çıkışa sahip olan hafıza elemanlarıdır. Burada $\overline{Q}$ çıkışının Q çıkışının tersi olması istenmektedir. Farklı giriş kombinasyonları için Latch'in Q çıkışının

- 1 yapılması (**set** edilmesi)
- 0 yapılması (**reset** edilmesi)
- Önceki durumu koruması (hafıza)
- Önceki durumu terslendirmesi

gibi işlemler yapması hedeflenir.

2.1.1 VEYADEĞİL Tabanlı SR Latch

VEYADEĞİL tabanlı SR Latch'in devresi Şekil 3 'de görüldüğü gibidir. Latch'in S ve R şeklinde iki girişi; yukarıda bahsetmiş olduğumuz üzere Q ve $\overline{Q}$ şeklinde iki çıkışı bulunmaktadır. Burada S girişi set işleminin baş harfidir. R ise tahmin edebileceğiniz üzere reset anlamına gelmektedir. Buna göre $S = 1$ yapıldığında çıkışın $Q = 1$ yapılması; $R = 1$ olduğunda ise çıkışın $Q = 0$ yapılması hedeflenmektedir. Diğer giriş kombinasyonlarının biri içinse latch'in hafıza olarak çalışması (önceki değeri tutması) beklenmektedir.

Şekil 3 'deki devrenin bunları yapıp yapmadığını belirlemek için devrenin analizini yapmamız gerekmektedir. Şekilden görüldüğü üzere iki tane geribesleme hattımız vardır. Bu iki geribesleme hattı bize (Şekil 2 'deki devrenin analizini yaparken bulmuş olduğumuz gibi) iki tane fark denklemi verecektir. Çıkışların mevcut değerlerini Q_n ve $\overline{Q}_n$ ile etiketlersek bu fark denklemlerini aşağıdaki gibi elde ederiz:

$$\begin{aligned} Q_{n+1} &= (R + \overline{Q}_n)' \\ \overline{Q}_{n+1} &= (S + Q_n)' \end{aligned} \tag{2}$$

Bu denklemleri kullanarak devreyi analiz etmeden önce $\overline{Q}$ çıkışının üstündeki sembolün DEĞİL işlemi olarak kullanılmadığını fark etmeniz önemlidir. Biz $\overline{Q}$ çıkışının Q çıkışının tersi olmasını istiyoruz ama bunun bir garantisi yok. Yani hesaplama yaparken $\overline{Q}$ çıkışı Q'nun tersi olmayabilir. Bunların tersleri sırasıyla $\overline{Q}'$ ve Q' ile gösterilecektir; $\overline{Q}' = Q$ olmak zorunda değildir.

SR Latch devresini analiz etmek için tek yapmamız gereken farklı giriş kombinasyonları için giriş değerlerini Eşitlik 2 'deki fark denklemlerinde yerine koymaktır. Sonrasında bulmuş olduğumuz sonraki durum değerlerini şimdiki durum yaparız ($Q_n = Q_{n+1}$ ve $\overline{Q}_n = \overline{Q}_{n+1}$). Ardından yeni Q_n ve $\overline{Q}_n$ değerlerini tekrar fark denklemlerine koyarak bir sonraki durumu hesaplarız. Bu işlemleri tekrar ederek çıkışın ne olacağını belirlemek mümkündür. Şimdi farklı giriş kombinasyonları için çıkışın ne olacağını belirleyelim:

$S = 1$, $R = 0$ **için**:

Çıkışların başlangıç durumları sırasıyla Q_0 ve $\overline{Q}_0$ olsun. Eşitlik 2 'de $S = 1$, $R = 0$ giriş değerlerini ve bu başlangıç durumlarını yerine koyarsak yukarıda bahsettiğimiz şekilde sırasıyla aşağıdaki durumları elde ederiz:

$$
\begin{array}{lll}
Q_{n+1} & = (0 + \overline{Q}_0)' & = \overline{Q}_0' \\
\overline{Q}_{n+1} & = (1 + Q_0)' & = 0 \\
& \Downarrow & \\
Q_{n+1} & = (0 + 0)' & = 1 \\
\overline{Q}_{n+1} & = (1 + \overline{Q}_0')' & = 0 \\
& \Downarrow & \\
Q_{n+1} & = (0 + 0)' & = 1 \\
\overline{Q}_{n+1} & = (1 + 1)' & = 0 \\
& \Downarrow & \\
& \vdots &
\end{array}
$$

Burada $\Downarrow$ sembolü sonraki durumu temsil etmektedir. İlk hesaplamada sonraki durumları $Q_{n+1} = \overline{Q}_0'$, $\overline{Q}_{n+1} = 0$ bulduğumuz için $\Downarrow$ sembolünden sonraki hesaplamada $Q_n = \overline{Q}_0'$ ve $\overline{Q}_n = 0$ alırız. Bulduğumuz bu yeni mevcut durumları ikinci hesaplamada yerine koyduğumuzda ise $Q_{n+1} = 1$ ve $\overline{Q}_{n+1} = 0$ elde ederiz. Bunları üçüncü hesaplamada $Q_n = 1$ ve $\overline{Q}_n = 0$ olarak kullandığımızda ise tekrar $Q_{n+1} = 1$, $\overline{Q}_{n+1} = 0$ bulduğumuz için sonraki hesaplamalarda sonuç değişmeyecek ve çıkışlar $Q = 1$ ve $\overline{Q} = 0$ için kararlı durumda olacaktır.

Buna göre, çıkış değerlerinin $S = 1$, $R = 0$ girişleri için sırasıyla alacağı değerler aşağıdaki gibi olacaktır:

$$\begin{pmatrix} Q \\ \overline{Q} \end{pmatrix} = \begin{pmatrix} Q_0 \\ \overline{Q}_0 \end{pmatrix} \rightarrow \begin{pmatrix} \overline{Q}_0' \\ 0 \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ 0 \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ 0 \end{pmatrix} \rightarrow \dots$$

Çıkışların $Q = 1$ ve $\overline{Q} = 0$ değerlerinde kararlı duruma geçmesi tam olarak set işleminde elde etmek istediğimiz sonuçtur.

$\underline{S = 0,\ R = 1}$ **için**:

Çıkışların başlangıç durumları sırasıyla Q_0 ve $\overline{Q}_0$ olsun. Eşitlik 2 'de $S = 0$, $R = 1$ giriş değerlerini ve bu başlangıç durumlarını yerine koyarsak sırasıyla aşağıdaki durumları elde ederiz:

$$\begin{aligned} Q_{n+1} &= (1 + \overline{Q}_0)' &&= 0 \\ \overline{Q}_{n+1} &= (0 + Q_0)' &&= Q_0' \\ &\Downarrow \\ Q_{n+1} &= (1 + Q_0')' &&= 0 \\ \overline{Q}_{n+1} &= (0 + 0)' &&= 1 \\ &\Downarrow \\ Q_{n+1} &= (1 + 1)' &&= 0 \\ \overline{Q}_{n+1} &= (0 + 0)' &&= 1 \\ &\Downarrow \\ &\vdots \end{aligned}$$

Buna göre, çıkış değerlerinin sırasıyla alacağı değerler aşağıdaki gibi olacaktır:

$$\begin{pmatrix} Q \\ \overline{Q} \end{pmatrix} = \begin{pmatrix} Q_0 \\ \overline{Q}_0 \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ Q_0' \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ 1 \end{pmatrix} \rightarrow \dots$$

Çıkışların $Q = 0$ ve $\overline{Q} = 1$ değerlerinde kararlı duruma geçmesi tam olarak reset işleminde elde etmek istediğimiz sonuçtur.

$\underline{S = 1,\ R = 1}$ **için**:

Bu giriş değerlerini Eşitlik 2 'de yerine koyduğumuzda Q_n ve $\overline{Q}_n$ değerleri ne olursa olsun aşağıdaki değerleri elde ederiz:

$$\begin{aligned} Q_{n+1} &= (1 + \overline{Q}_n)' = 0 \\ \overline{Q}_{n+1} &= (1 + Q_n)' = 0 \end{aligned}$$

Buna göre, $S = 1$, $R = 1$ olduğu müddetçe çıkışlar $Q = \overline{Q} = 0$ olacaktır. Normalde bizim $\overline{Q}$ çıkışı için istediğimiz Q çıkışının tersi olmasıydı. Bu şart bu girişler için sağlanmamaktadır.

$S = 0$, $R = 0$ **için**:

Çıkışların başlangıç durumları sırasıyla Q_0 ve $\overline{Q}_0$ olsun. Eşitlik 2 'de $S = 0$, $R = 0$ giriş değerlerini ve bu başlangıç durumlarını yerine koyarsak sırasıyla aşağıdaki durumları elde ederiz:

$$\begin{aligned}
Q_{n+1} &= (0 + \overline{Q}_0)' &&= \overline{Q}_0' \\
\overline{Q}_{n+1} &= (0 + Q_0)' &&= Q_0' \\
&\Downarrow \\
Q_{n+1} &= (0 + Q_0')' &&= Q_0 \\
\overline{Q}_{n+1} &= (0 + \overline{Q}_0')' &&= \overline{Q}_0 \\
&\Downarrow \\
Q_{n+1} &= (0 + \overline{Q}_0)' &&= \overline{Q}_0' \\
\overline{Q}_{n+1} &= (0 + Q_0)' &&= Q_0' \\
&\Downarrow \\
&\vdots
\end{aligned}$$

Buna göre, çıkış değerleri aşağıdaki değerler arasında osilasyon yapacaktır:

$$\begin{pmatrix} Q \\ \overline{Q} \end{pmatrix} = \begin{pmatrix} Q_0 \\ \overline{Q}_0 \end{pmatrix} \to \begin{pmatrix} \overline{Q}_0' \\ Q_0' \end{pmatrix} \to \begin{pmatrix} Q_0 \\ \overline{Q}_0 \end{pmatrix} \to \begin{pmatrix} \overline{Q}_0' \\ Q_0' \end{pmatrix} \to \dots$$

Buna göre farklı başlangıç durumları için farklı sonuçlar elde ederiz. Eğer $Q_0 = 0$, $\overline{Q}_0 = 1$ olursa SR Latch çıkışları aşağıdaki gibi olacaktır:

$$\begin{pmatrix} Q \\ \overline{Q} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \to \begin{pmatrix} 0 \\ 1 \end{pmatrix} \to \begin{pmatrix} 0 \\ 1 \end{pmatrix} \to \dots$$

Görüldüğü üzere $Q_0 = 0$, $\overline{Q}_0 = 1$ için sonraki durumlarda başlangıç durumu kararlı bir şekilde korunmaktadır.

***

Eğer $Q_0 = 1$, $\overline{Q}_0 = 0$ olursa SR Latch çıkışları aşağıdaki gibi olacaktır:

$$\begin{pmatrix} Q \\ \overline{Q} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \to \begin{pmatrix} 1 \\ 0 \end{pmatrix} \to \begin{pmatrix} 1 \\ 0 \end{pmatrix} \to \dots$$

$Q_0 = 0$, $\overline{Q}_0 = 1$ için de sonraki durumlarda başlangıç durumu kararlı bir şekilde korunmaktadır.

***

S	R	Q_{n+1}	$\overline{Q}_{n+1}$	
0	0	Önceki Durum		
0	1	0	1	
1	0	1	0	
1	1	0	0	(Yasaklı Durum)

Tablo 1: VEYADEĞİL tabanlı SR Latch'in doğruluk tablosu.

Bir önceki durumda görmüş olduğumuz üzere $S = R = 1$ yaptığımızda çıkışları $Q = \overline{Q} = 0$ yapabiliyorduk. Buna göre, girişleri $S = 0$, $R = 0$ yaptığımızda başlangıç şartları $Q_0 = \overline{Q}_0 = 0$ olabilir. Bu durumda sonraki çıkışlar aşağıdaki gibi olacaktır:

$$\begin{pmatrix} Q \\ \overline{Q} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} 0 \\ 0 \end{pmatrix} \rightarrow \begin{pmatrix} 1 \\ 1 \end{pmatrix} \rightarrow \dots$$

Pratikte bu durumda çıkışlar sürekli osilasyon yapmayacaktır. Bir süre bu şekilde osilasyon yapıldıktan sonra devrenin geribesleme hatlarından birinde yaşanacak en ufak bir gecikme $(Q_n, \overline{Q}_n)$ çıkışlarının $(0, 1)$ ve $(1, 0)$ şeklindeki iki kararlı durumdan birine düşmesine neden olup, çıkışlar bu durumlardan birinde sabit kalacaktır. Fakat, latch'in hangi kararlı duruma düşeceği (hangi hattın geç kalacağı) belirsizdir. Hafıza elemanının bu davranışı **metakararlılık** olarak adlandırılır. Dijital devrelerde meydana gelecek bir metakararlılık sistemin hatalı çalışmasına neden olacağı için istenmeyen bir durumdur.

***

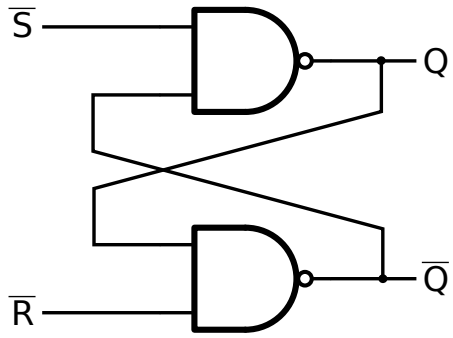
Bu sonuçlardan görmüş olduğumuz üzere $S = 0$, $R = 0$ için, çıkışların başlangıç durumu $Q = \overline{Q} = 0$ olmadığı müddetçe SR Latch önceki durumu korur.

Girişlerin $S = 1$, $R = 1$ yapılması

1. Q ve $\overline{Q}$ değerlerinin istemediğimiz şekilde aynı değeri almasına;
2. Bu durumdayken $S = 0$, $R = 0$ yapılırsa latch'in metakararlılığa sürüklenmesine

neden olacağından, SR Latch'in girişlerinin $S = 1$, $R = 1$ yapılması istenmez. Bu giriş değerleri doğruluk tablosunda **istenmeyen** veya **yasaklı** durum olarak adlandırılır.

Elde ettiğimiz bu sonuçlarla VEYADEĞİL tabanlı SR Latch'in doğruluk tablosu Tablo 1 'deki gibi elde edilir.



Şekil 4: VEDEĞİL tabanlı SR Latch.

2.1.2 VEDEĞİL Tabanlı SR Latch

VEDEĞİL tabanlı SR Latch'in devresi Şekil 4 'deki gibidir. Çalışması VEYADEĞİL tabanlı SR Latch'in tam tersidir. Bu nedenle girişleri $\overline{S}$ ve $\overline{R}$ ile ifade edilir ve bazen negatif SR Latch olarak da adlandırılır. Bu devre için durum denklemleri aşağıdaki gibi elde edilir:

$$\begin{aligned} Q_{n+1} &= (\overline{S}.\overline{Q}_n)' \\ \overline{Q}_{n+1} &= (\overline{R}.Q_n)' \end{aligned} \tag{3}$$

Bu denklemleri kullanarak yukarıdaki gibi bir analiz yaptığımızda Tablo 2 'deki gibi bir doğruluk tablosu elde ederiz. Bu devrenin analizi size bırakılmıştır.

$\overline{S}$	$\overline{R}$	Q_{n+1}	$\overline{Q}_{n+1}$	
0	0	1	1	(Yasaklı Durum)
0	1	1	0	
1	0	0	1	
1	1	Önceki Durum		

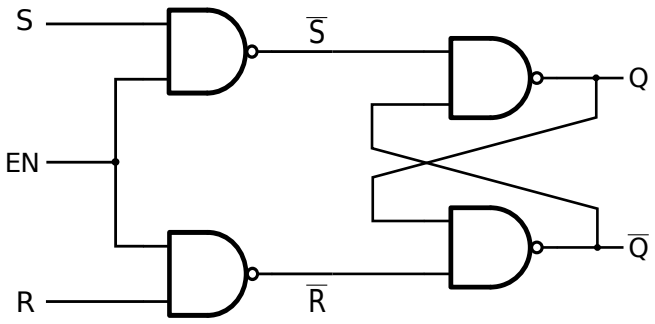
Tablo 2: VEDEĞİL tabanlı SR Latch'in doğruluk tablosu.

2.1.3 Enable Girişli SR Latch

VEDEĞİL tabanlı SR Latch'i Şekil 5 'deki gibi modifiye edersek latch'e enable girişi eklemiş oluruz. Şekilden görüldüğü üzere $EN = 0$ olduğunda S ve R girişleri ne olursa olsun devrenin içindeki $\overline{S}$ ve $\overline{R}$ hatları 1 olacağından Tablo 2'e göre latch önceki durumu korur. $EN = 1$ olduğunda ise devrenin içindeki $\overline{S}$ ve $\overline{R}$ hatları sırasıyla S ve R girişlerinin tersi olacaktır. Bu durumda SR Latch terslendirilmiş haliyle değil olması gerektiği gibi çalışacaktır. Dolayısıyla, devrenin doğruluk tablosu Tablo 3 'deki gibi olacaktır.

EN	S	R	Q_{n+1}	$\overline{Q}_{n+1}$	
0	X	X	Önceki Durum		
1	0	0	Önceki Durum		
1	0	1	0	1	(Reset)
1	1	0	1	0	(Set)
1	1	1	0	0	(Yasaklı Durum)

Tablo 3: Enable girişli SR Latch'in doğruluk tablosu.



Şekil 5: Enable girişli SR Latch devresi.

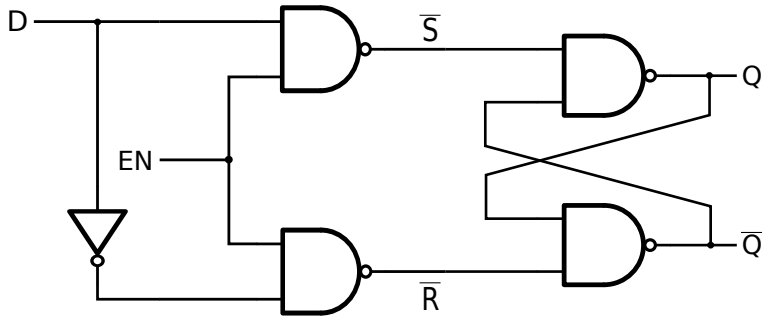
Buna göre $EN = 1$ olduğu müddetçe S ve R girişlerinde yapılan değişikliklere göre latch güncellenecektir. Fakat, $EN = 0$ olursa latch S ve R girişlerindeki değişimlere bakmadan önceki durumu korumaya devam ederek deaktive olacaktır.

2.1.4 D Latch

SR Latch'in $S = 1, R = 1$ istenmeyen durumunun girilmemesi tamamen tasarımcının kontrolündedir. Fakat, yine de her iki giriş de yanlışlıkla 1 yapılabilir. Bu durumda devre istenilmeyen şekilde çalışacak ve dijital sistemde hatalar meydana gelecektir. SR Latch için istenmeyen durumun asla girilmemesi için tek bir girişi S'ye, DEĞİL kapısı ile terslendirilmiş halini de R'ye bağlayabiliriz. Böylelikle, SR Latch sadece set ve reset durumları için çalışacak; enable girişi ile de latch'in önceki durumu tutması sağlanacaktır. Bu düzenlemede latch aktifken giriş 1 olursa çıkış set edileceği; giriş 0 ise çıkış reset edileceğinden, bu latch için giriş neyse bir sonraki çıkış değeri girişin aynısı olacaktır. Anlatılan bu devre düzenlemesi Şekil 6 'deki gibi olup bu latch düzenlemesine D Latch adı verilir. Burada D "data"nın kısaltmasıdır, çünkü D Latch etkinken girişteki datanın (verinin) tutulmasını sağlar.

Buna göre, D Latch'in doğruluk tablosu Tablo 4 'deki gibi olacaktır.

Şimdiye kadar görmüş olduğumuz latch'lerin blok diyagramları Şekil 7 'de verilmektedir.



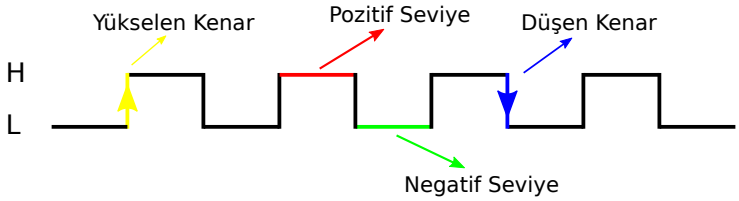
Şekil 6: D Latch devresi.

EN	D	Q_{n+1}	$\overline{Q}_{n+1}$
0	X	Önceki Durum	
1	0	0	1
1	1	1	0

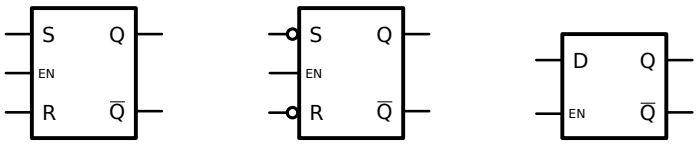
Tablo 4: D Latch'in doğruluk tablosu.

2.2 Flip Floplar

Daha önce söylemiş olduğumuz üzere flip floplar latch'lerin senkron hale getirilmiş halleri olup, senkron ardışıl devre tasarımında kullanılırlar. Senkronizasyon için kullanılan periyodik saat (clock) darbeleri aşağıdaki gibidir:

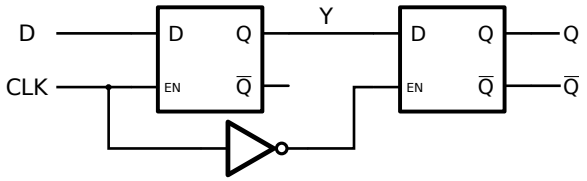


Latch'lerin enable girişine bu saat darbelerini uyguladığımızda latch'ler pozitif seviyede güncellenir. Eğer enable girişini değil kapısı ile terslendirirsek, bu durumda



(a) (b) (c)

Şekil 7: (a) VEYADEĞİL tabanlı SR Latch blok diyagramı; (a) VEDEĞİL tabanlı SR Latch blok diyagramı; (c) D Latch blok diyagramı.



Şekil 8: Düşen kenar tetiklemeli D flip flop master slave gerçekleştirimi.

latch negatif seviyede güncellenir. Fakat, bu bize kısmi senkronizasyon sağlasa da tam anlamıyla senkronizasyon elde edemeyiz, çünkü senkronizasyon sinyali pozitif (veya negatif) seviyede kaldığı müddetçe latch'ler güncellenmeye devam edecektir. Bu nedenle, bu süre içerisinde yine farklı hafıza elemanları farklı zamanlarda güncellenebilir.

Bu nedenle, tam anlamıyla senkronizasyon sağlayabilmek için hafıza elemanlarının sadece bir tek bir anda topluca güncellenmesi gerekir. Saat darbelerine baktığımızda darbelerin yükselen veya düşen kenarları için yapılacak güncellemelerin bu tür anlık güncellemelere gayet uygun olduğunu görebiliriz. İşte flip floplar saat darbelerinin yükselen veya düşen kenarları için "anlık" olarak güncellenen hafıza elemanlarıdır.

2.2.1 D Flip Flop

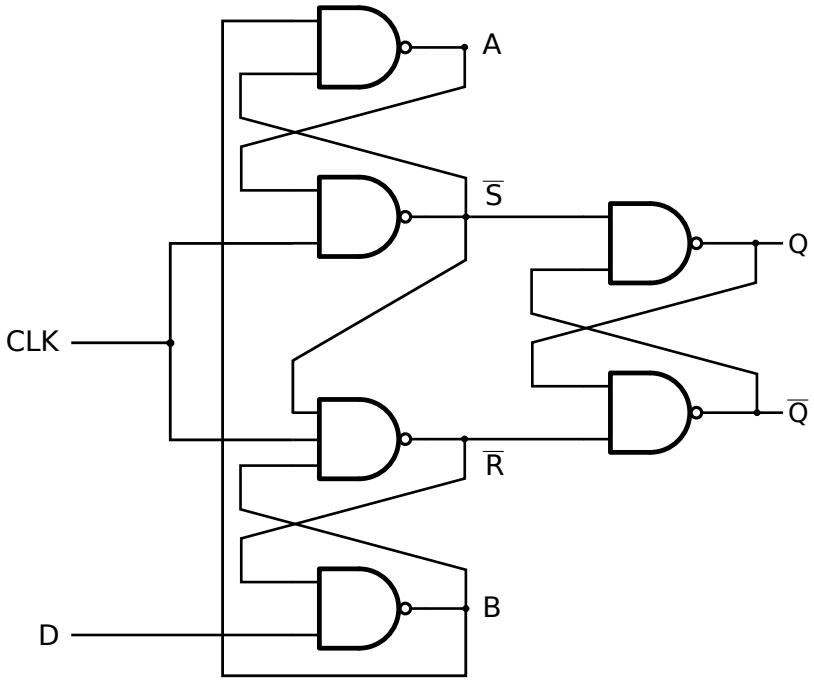
Kenar tetiklemeli D flip flop elde etmek için latch'ler kullanılır. Bu tür bir flip flop üretmenin ilk yolu iki tane D latch alıp bunu Şekil 8 'deki gibi "master slave" bağlantısı yapmaktır. Bu tür bağlantıda birinci (master) latch çıkışı ikinci (slave) latch girişine bağlanır. CLK girişi üzerinden verilen saat darbeleri ise latch'lerin enable girişine uygulanırken ikinci latch'e terslendirilerek verilir.

Bu devrenin çalışmasını kavrayabilmek için saat darbesi pozitif seviyedeyken ve negatif seviyedeyken neler olduğunu incelememiz lazım.

Pozitif Seviye (H): Birinci (master) latch aktif olduğu için saat darbesi H seviyesinde olduğu müddetçe D girişi Y hattına aktarılır. Bu sırada ikinci (slave) latch aktif değildir ve önceki durumunu korur.

Negatif Seviye (L): Birinci (master) latch aktif değildir; Y hattında saat darbesi L seviyesine düşmeden hemen önce aktarılmış olan D giriş değeri vardır. İkinci latch (slave) ise aktif olduğundan Q çıkışına bu Y hattının değeri aktarılır. Birinci latch aktif olmadığından Y hattındaki değer değişemez ve L seviyesi boyunca ikinci latch aktif olsa da çıkışa sürekli sabit Y değerini aktaracaktır.

Bu iki durumu birleştirdiğimizde, her saat darbesi periyodunda saat darbesi H seviyesinden L seviyesine düşmeden hemen önceki D giriş değerinin çıkışa aktarılarak



Şekil 9: Pozitif kenar tetiklemeli klasik D flip flop gerçekleştirimi.

bir sonraki düşen kenara kadar tutulduğunu fark ederiz. Bir sonraki düşen kenarda yine aynı şeyler olur ve çıkış güncellenir. Bu nedenle, Şekil 8 'deki D latch konfigürasyonu düşen kenar tetiklemeli D flip flop olarak adlandırılır. Bu flip flop latch'lerin aksine her düşen kenarda güncellenir. Bu flip flop ile saat darbelerinin sadece düşen kenarlarında güncellenen senkron devreler oluşturmak mümkündür.

Master slave konfigürasyonu ile yükselen kenarda tetiklenen D flip flop elde etmek için tek yapmamız gereken Şekil 8 'de ikinci (slave) latch enable girişini terslendirmek yerine, birinci (master) enable girişini terslendirmektir. Bu bağlantıyı oluşturarak yükselen kenarda tetiklenip tetiklenmediğini kontrol edin.

Şekil 6 'den tek bir D latch oluşturmak için 4 adet VEDEĞİL ve 1 adet DEĞİL kapısına ihtiyacımız olduğunu görürüz. Buna göre, yukarıda bahsettiğimiz master slave D flip flopu gerçekleştirmek için toplam 8 VEDEĞİL kapısına ve 2 tane DEĞİL kapısına ihtiyacımız vardır. Fakat, D flip flopun ticari uygulamalarında (mesela 74 serisinde) sadece 6 VEDEĞİL kapısı ile gerçekleştirebileceğimiz "klasik D flip flop devresi" kullanılır. Klasik D flip flop devresi Şekil 9 'de görüldüğü gibi üç tane SR Latch'in birleştirilmesiyle oluşturulmuştur.

Fark edeceğiniz üzere çıkışta VEDEĞİL tabanlı bir SR latch olduğu için $\overline{S}$ ve $\overline{R}$ hatlarındaki değerlerin ne olacağını belirlemek devreyi analiz etmemize yetecektir. Bunun için devredeki A ve B değişkenlerinden kurtularak $\overline{S}$ ve $\overline{R}$ durum denklemlerini CLK ve D cinsinden oluşturmamız gerekir. Buna göre,

(a) (b)

Şekil 10: (a) Pozitif kenar tetiklemeli D flip flopun; (b) negatif kenar tetiklemeli D flip flopun blok diyagramı.

$$\overline{S}_{n+1} = (CLK.A)' = CLK' + A' = CLK' + B.\overline{S}_n = CLK' + \overline{S}_n(D' + \overline{R}_n)$$

$$\overline{R}_{n+1} = (CLK.\overline{S}_n.B)' = CLK' + \overline{S}'_n + D.\overline{R}_n$$

Klasik D flip flop devresinin asenkron analizinden sorumlu değilsiniz. Bu sadece bilgi amaçlı verilmektedir.

eşitliklerini elde ederiz. Bu eşitlikleri kullanarak D ve CLK girişlerinin 4 farklı kombinasyonu için aşağıdaki sonuçları elde ederiz:

$CLK = 0$ $D = 0$	için	$\overline{S}_{n+1} = 1$ $\overline{R}_{n+1} = 1$
$CLK = 0$ $D = 1$	için	$\overline{S}_{n+1} = 1$ $\overline{R}_{n+1} = 1$
$CLK = 1$ $D = 0$	için	$\overline{S}_{n+1} = \overline{S}_n$ $\overline{R}_{n+1} = \overline{S}'_n$
$CLK = 1$ $D = 1$	için	$\overline{S}_{n+1} = \overline{S}_n\overline{R}'_n$ $\overline{R}_{n+1} = \overline{S}'_n + \overline{R}_n$

Bu sonuçlara göre $CLK = 0$ olduğunda D değeri ne olursa olsun $\overline{S}$ ve $\overline{R}$ hatları 1 olmaktadır. Bu ise çıkıştaki latch'in çıkışlarının değişmeyeceği anlamına gelir. $CLK = 1$ olur olmaz $\overline{S}_n = \overline{R}_n = 1$ olacağından $D = 0$ ise $\overline{S} = 1$, $\overline{R} = 0$; $D = 1$ ise $\overline{S} = 0$, $\overline{R} = 1$ değerleri elde edilir. Yani $D = 0$ için çıkış latch'i reset edilirken, $D = 1$ için set edilir. Bir kez $\overline{S}_n = 1$, $\overline{R}_n = 0$ (veya tam tersi) olursa $CLK = 1$ iken D girişi ne kadar değişirse değişsin, latch'in çıkışı tam olarak $CLK = 0$'dan $CLK = 1$'e geçiş anındaki D değeri olacaktır. Bu ise pozitif kenar tetikleme anlamına gelir. Dolayısıyla, Şekil 9 'deki devre pozitif kenar tetiklemeli bir D flip flop devresidir.

Pozitif ve negatif kenar tetiklemeli D flip flopun blok diyagramları Şekil 10 'de görüldüğü gibidir.

2.2.2 JK Flip Flop

Bu bölümün başında bir hafıza elemanının önceki durumun terslendirebileceğini söylemiştik. Görmüş olduğumuz ne SR Latch ne de D flip flop bu özelliği sağlamıyordu. JK flip flop ise J ve K girişlerinin değerine bağlı olarak bu işlemi de yapabilen bir flip floptur. Çalışması SR Latch'e benzerdir. J girişi çıkışı set etmek için; K girişi çıkışı reset etmek için kullanılır. Her iki giriş sıfır olursa tıpkı SR Latch'de olduğu gibi önceki değer korunur. SR Latch'den farklı olarak $J = K = 1$ yasaklı durum değildir ve önceki değeri terslendirmek için kullanılır. Buna göre JK flip flopun doğruluk tablosu Tablo 5 'deki gibi olacaktır.

J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	Q'_n

Tablo 5: JK flip flopun doğruluk tablosu.

Mümkün olan en az mantık kapısı kullanılarak gerçekleştirilebilmesi ve pratik olması bakımından dijital devre tasarımında en yaygın kullanılan flip flop D flip floptur. Diğer daha az kullanılan flip floplar sıklıkla D flip floplar kullanılarak gerçekleştirilir. Bu bakımdan JK flip flop da D flip flop ve mantık kapıları kullanılarak kolaylıkla gerçekleştirilebilir. Bunun için

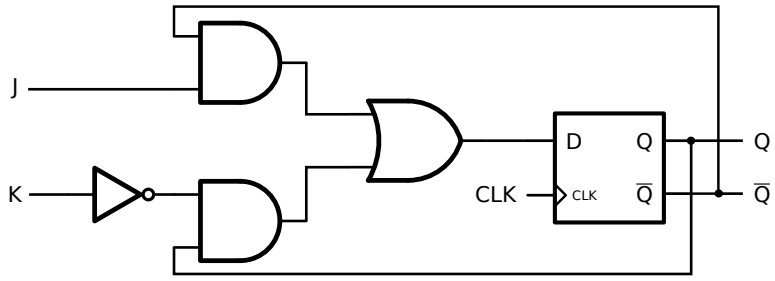
$$D = JQ' + K'Q$$

eşitliğini inceleyelim. Burada D girişi D flip flop girişidir. Eğer $J = K = 0$ olursa $D = Q$ olacağı için flip flop çıkışında önceki değer korunur. $J = 0$, $K = 1$ için $D = 0$ yani çıkış reset edilir. $J = 1$, $K = 0$ için $D = 1$ olduğundan çıkış set edilir. Son olarak $J = K = 1$ için $D = Q'$ bulunur. Yani önceki değer terslendirilir. Bu bakımdan, D girişine bu eşitlikteki kombinasyonel devre bağlantılarını yaparsak JK flip flopun çalışmasının aynısını elde ederiz. Buna göre, JK flip flopun D flip flop kullanılarak yapılan gerçekleştirimi Şekil 11 'deki gibi olacaktır.

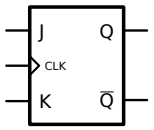
JK flip flopun blok diyagramı ise Şekil 12 'de görüldüğü gibidir.

2.2.3 T Flip Flop

T flip floptaki T toggle'dan (terslendirme) gelmektedir. $T = 0$ için önceki durum korunurken, $T = 1$ için önceki durum terslendirilir. Önceki durumun terslendirilmesinden T flip flop ile JK flip flop arasında bir ilişki olduğunun farkına varmanız lazım. Aslına bakarsanız, T flip floptaki $T = 0$ durumu JK flip floptaki $J = K = 0$ durumuna; T flip floptaki $T = 1$ durumu da JK flip floptaki $J = K = 1$ durumuna



Şekil 11: JK flip fopun D flip flop kullanılarak gerçekleştirimi.



Şekil 12: JK flip flopun blok diyagramı.

karşılık gelir. Buradan T flip flopun, JK flip flopun J ve K girişlerinin birbirine bağlanmasıyla elde edilebileceğini anlarız. Buna göre T flip flopun doğruluk tablosu Tablo 6 'daki gibi olacaktır.

T	Q_{n+1}
0	Q_n
1	Q'_n

Tablo 6: T flip flopun doğruluk tablosu.

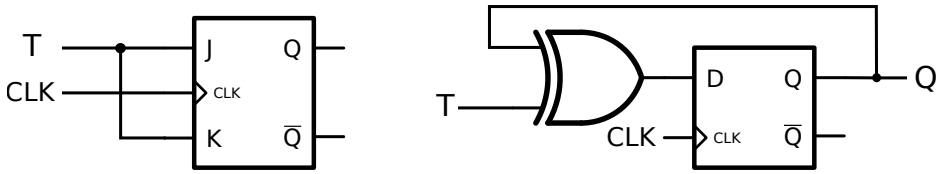
T flip flopu D flip flop kullanarak da gerçekleştirebiliriz. Bunun için $T = 0$ olduğunda $D = Q$ ve $T = 1$ olduğunda $D = Q'$ elde etmemiz gerekir. Bunu

$$D = T'Q + TQ' = T \oplus Q$$

denklemiyle kolayca sağlayabiliriz. Dolayısıyla, T flip flopun JK ve D flip floplar kullanılarak gerçekleştirimi Şekil 13 'deki gibi olacaktır. T flip flopun blok diyagramı ise Şekil 14 'deki gibidir.

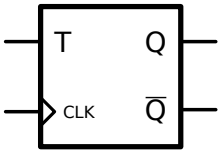
2.2.4 Asenkron Girişler

Flip flopların pratik kullanımında flip floplara enerji verildiğinde flip flopların bulunduğu durumların ne olacağı çoğu zaman bilinmez. Dolayısıyla, flip floplar saat darbesi ile güncellenmeden hemen önce flip flopları istenilen başlangıç durumuna sokmak için asenkron kontrol girişleri kullanılır. Asenkron kontrol girişleri saat darbesini beklemeksizin flip flop çıkışını belli bir değere ayarlar. Eğer çıkışı asenkron bir şekilde 1 yapan bir giriş varsa bu giriş **preset** olarak adlandırılır. Tam tersi çıkışı 0 yapan asenkron giriş **clear** olarak adlandırılır.



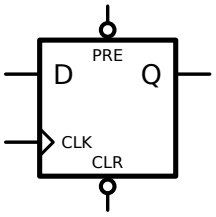
(a) (b)

Şekil 13: T flip flopun (a) JK flip flop; (b) D flip flop ile gerçekleştirimi.



Şekil 14: T flip flopun blok diyagramı.

Aktif düşük preset girişine ve aktif düşük clear girişine sahip bir D flip flopun blok diyagramı Şekil 15 'de görüldüğü gibidir. Bir önceki notlardan hatırlayacağınız üzere bu girişlerin aktif düşük olması 0 değeri için yapmaları gereken işlevleri yerine getireceklerini belirtir. Buna göre, bu flip flopun preset girişi 0 yapılırsa, saat darbesi güncellemesi beklenmeden flip flop çıkışı 1 yapılır. Clear 0 yapılırsa bu sefer çıkış asenkron bir şekilde 0 yapılır.



Şekil 15: Asenkron girişli D flip flop.