

EEM212 - SAYISAL DEVRE TASARIMI DERS NOTLARI

DERS NOTU 7: ARDIŞIL DEVRELER III: TASARIM

Dr. İsmail Öztürk *

<ismail.ozturk@amasya.edu.tr>

(ver. 1.1)

İçindekiler

1 Senkron Ardışıl Devre Tasarımı	1
1.1 Durum İndirgemesi	4
1.2 D FF ile Tasarım	6
1.3 JK FF ile Tasarım	9
1.4 T FF ile Tasarım	14
2 FF Karakteristik Denklemlerinin Elde Edilmesi	17

1 Senkron Ardışıl Devre Tasarımı

Devre tasarlanırken önce elimizdeki problemin ne olduğunu incelememiz lazım. Ardışıl devre tasarımı eğer elimizdeki problem ardışıklık veya hafıza gerektiriyorsa yapmamız gerekir. Aksi takdirde daha önce görmüş olduğumuz gibi kombinasyonel devre tasarlamak elimizdeki problemi çözmeye yetecektir.

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

Elimizdeki problemin ardışılık içerip içermediğini problemi önceki durum / sonraki durum ilişkisi içerisinde ifade edip edemeyeceğimizi belirleyerek anlarız. Eğer problem önceki durum / sonraki durum ilişkisi ile ifade edilebiliyorsa o problemi sonlu durum makinesi ile ifade edebiliriz. Dolayısıyla, eğer elimizdeki problem bir sonlu durum makinesi ile ifade edilebiliyorsa o problemin ardışılık içerdiğini ve ardışıl devre ile gerçekleştirilebileceğini anlarız.

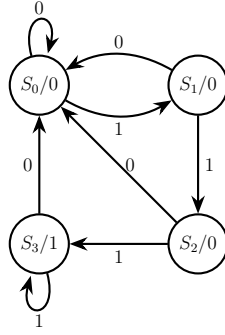
Hatırlayacağınız üzere, bir problemi sonlu durum makinesi olarak ifade etmek için yapmamız gereken bir durum diyagramı oluşturmaktır. Bu nedenle, tasarımda yapmamız gereken ilk şey elimizdeki probleme ait durum diyagramını oluşturmak olmalıdır. Bunun için öncelikle problemin her bir durumuna bir etiket vermeniz gerekir (S_0 , S_1 , a , b gibi). Daha sonra elimizdeki problemin gereksinimlerine ve tasarım tercihinize göre sonlu durum makinesini oluşturmak için Mealy mi yoksa Moore tasarımı mı yapacağınızı belirlemeniz gerekir. Bunu belirledikten sonra Mealy veya Moore tasarımına uygun olarak durum diyagramı oluşturulur.

Örnek olarak elimizdeki problem “bir seri veri yolunda iletilen bitlerde art arda üç veya daha fazla 1 gelmesini belirlemek” olsun. Yani, elimizde 001010110... gibi seri olarak iletilen bir veri var ve biz bu veri içerisinde 111, 1111, 11111... gibi paternlerin bulunup bulunmadığını belirlemeye çalışıyoruz. Bu problem hafıza içerdiği için ardışıl devre tasarımıyla çözümlenmelidir. Bu probleme karşılık gelen sonlu durum makinesinin bir girişi ve bir çıkışı olmalı. Giriş değeri iletilen bit dizisinden alınırken, çıkış yalnızca girişten alınan bit değerleri art arda üç veya daha fazla kez 1 olduğunda 1 olmalı. Diğer tüm durumlarda çıkış 0 olmalı.

Bu probleme karşılık gelen sonlu durum makinesinin ilk durumu “**başlangıç durumu**” olarak seçilir. Yani kurduğumuz devre ilk çalıştırıldığında veya resetlendiğinde bu duruma geçmelidir. Bu durumu S_0 ile etiketleyelim. Başlangıç durumunda sonlu durum makinesinin girişi iletilen bit dizisinden alınan bir biti incelemeli. Eğer giriş 0 ise istenilen şart sağlanmadığından (art arda 1 gelmesi), sonraki durumda da başlangıç durumunda kalınmalıdır. Eğer giriş 1 ise S_1 ile etiketleyeceğimiz sonraki duruma geçiş yapılmalı. S_0 durumunda art arda 1 gelme şartı sağlanmadığından çıkış 0 (mantıksal yanlış) yapılmalı.

S_1 durumundayken veri dizisinde iletilen bir önceki bit değeri 1 olmak zorunda. Eğer bu durumdayken giriş 0 olursa, iletilen bit dizisi 10 olacağından istediğimiz şart sağlanmaz ve tekrar başlangıç durumu olan S_0 durumuna geri dönmemiz gerekir. Tam tersi giriş 1 olursa, iletilen bit dizisi 11 olacağı için sonraki durum olan S_2 durumuna geçmemiz gerekir. Bu durumda da iletilen bit dizisi 111 veya 1111 gibi bir değer olmadığı için çıkışımız yine 0 olmalı.

S_2 durumundayken veri dizisinde iletilen önceki bit değerleri 11 olmak zorunda. Bu durumda, giriş 0 ise, iletilen bit dizisi 110 olacağından istediğimiz şart sağlanmaz ve tekrar başlangıç durumuna dönerek patern aramaya baştan başlamamız gerekir. Eğer, giriş 1 olursa, iletilen bit dizisi 111 olacağından bu durumu ele alacak olan sonraki durum olan S_3 durumuna geçiş yapmamız gerekir. Bu durumdayken de henüz istediğimiz şart kesin olarak sağlanmadığından (iletilen veri 110 veya 111 olabilir), çıkış 0 yapılmalıdır.



Şekil 1: Patern dedektörünün Moore tipi durum diyagramı.

S_3 durumunda veri disisinde iletilen önceki bit değerleri kesin olarak 111 olmak zorunda. Bu nedenle, S_3 durumundaki çıkış 1 yapılmalı (çünkü art arda en az üç kez 1 gelme şartı sağlanmış oldu). Eğer giriş bu durumdayken de 1 ise, sonraki durumda iletim 1111 olacağından yine istediğimiz şart sağlanır. Dolayısıyla, 1 girişi için sonraki durumda da S_3 durumunda kalabiliriz. Giriş tekrar 1 olursa iletim 11111 olur yine şart sağlanır. Gördüğünüz üzere giriş 1 olduğu müddetçe S_3 durumunda kalabiliriz ve çıkış şartın sağlandığını belirtecek şekilde 1 (mantıksal doğru) olmalı. Eğer giriş 0 olursa, art arda 1 gelme şartı bozulacağından sonraki durumda tekrar S_0 durumuna dönerek patern arayışına baştan başlamamız gerekir.

Şimdi yukarıda anlattığımız çalışma prensibini durum diyagramına aktaralım. Yukarıda sonlu durum makinesinin çıkışının sadece durumlara bağlı olduğu açıkça görülmektedir. Bu nedenle, tasarımı Moore makinesi olmalıdır. Belirlemiş olduğumuz durum etiketlerini kullanıp yukarıdaki ilişkileri oklarla ifade ettiğimizde Moore tipi durum diyagramı Şekil 1'deki gibi elde edilecektir.

Durum diyagramını elde ettikten sonra yapabiliyorsak durum indirgemesi ¹ yapmamız gerekir. Durum indirgemesi, oluşturmuş olduğumuz durumlardan gereksiz yere oluşturulmuş olanlar varsa bunları tespit edip eleyerek kuracağımız devreyi daha da basitleştirir. Bunu nasıl yapacağımızı bir sonraki kısımda göreceğiz.

Durum indirgemesi yaptıktan sonra artık durum diyagramı ile sonlu durum makinesine karşılık gelen dijital devreyi oluşturmaya başlayabiliriz. Bunun için öncelikle yapmamız gereken durum tablosunu oluşturmaktır. Bir önceki ders notundan hatırlayacağınız üzere durum tablosundaki durumlar 00, 01, 101, 111 gibi bitlerle ifade edilmekteydi. Dolayısıyla, durum tablosunu oluşturmak için S_0, S_1, a, b gibi değerlerle etiketlediğimiz durumlara ikili kodlar atamalıyız. Durumlara bu kodları atarken, n bitlik bir kodlamanın 2^n farklı durumu ifade edebileceğini göz önünde bulunduramamız gerekir. Yani, 4 farklı durumumuz varsa bunu 2-bitlik 00, 01, 10, 11 değerleriyle kodlamamız gerekir. Benzer şekilde, 4 ile 8 arasındaki durumları ifade etmek için 3-bitlik 000, 001, ... şeklinde bir kodlama yapmamız gerekir.

Durum tablosunu oluşturduktan sonra kullanacağımız flip flop tipini belirlememiz gerekir. Flip flop tipini belirledikten sonra yapmamız gereken flip flop girişlerine

¹ing. *state reduction*

bağlanacak olan kombinasyonel devreler ile çıkış devrelerinin Boole eşitliklerini belirlemektir. Bunun için, daha önce görmüş olduğumuz kombinasyonel devre tasarımını kullanırız. **Bu aşamada, her zaman için devreyi en sade haliyle oluşturmamız gerekir.** Yani, kombinasyonel devreleri mantık kapıları ile oluştururken Karnaugh haritaları kullanmalıyız.

Gerekli kombinasyonel devrelerin eşitliklerini belirledikten sonra son olarak yapmamız gerekenler devre şematüğini çizmek ve devrenin doğru çalışıp çalışmadığını görmek için simülasyon yapmaktır.

Şimdiye kadar anlatmış olduğumuz tasarım aşamalarını aşağıdaki gibi özetleyebiliriz:

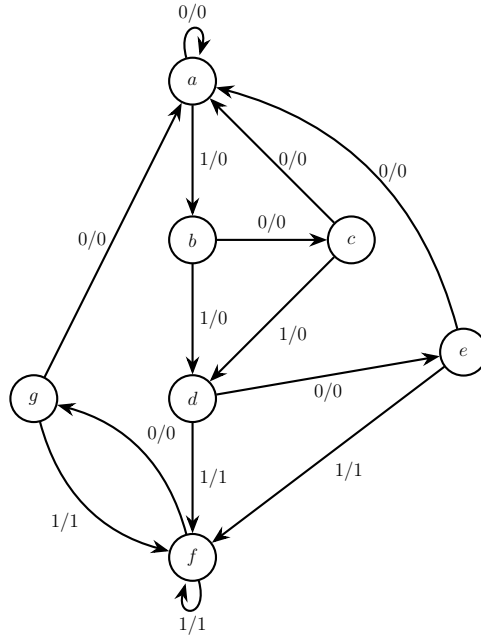
1. Durumları belirleyerek durum diyagramını oluşturmak.
2. Gerekirse durum indirgemesi yapmak.
3. Her bir duruma ikili kod verdikten sonra durum tablosunu oluşturmak.
4. Kullanılacak flip flop türünü seçmek.
5. Flip flop girişlerine bağlanacak kombinasyonel devreler ile çıkışlara ait devrelerin Boole eşitliklerini bulmak.
6. Devre şematüğünü çizmek ve devrenin simülasyonunu yapmak.

1.1 Durum İndirgemesi

Bazen oluşturmuş olduğumuz sonlu durum makinesinde bazı durumlar aynı giriş değerleri için aynı çıkışları üretip aynı sonraki duruma sahip olabilirler. Bu şekilde bir ilişkiye sahip durumlar eşdeğer durumlar olarak adlandırılır ve bu durumlardan sadece birini kullanmak da aynı işlevi sağlayacaktır. Bu sayede aynı sonlu durum makinesini daha az durum ile ifade etmiş oluruz. tasarımda daha az durum kullanmaksa, devreyi gerçekleştirirken daha az flip flop kullanmanızı sağlayacaktır. İşte yapılan bu durum azaltma işlemine durum indirgemesi adı verilir.

Durum indirgemesi yapmak çok basit bir işlemdir. Tek yapmanız gereken durumları bir durum tablosunda alt alta yazarak aynı satırlara sahip durumları belirlemektir. İşte tabloda bu aynı satırlara sahip olan durumlar eşdeğer durumlardır ve bunlardan sadece birini kullanmak yeterli olacaktır. Örnek olarak Şekil 2'deki durum diyagramını inceleyelim.

Durum diyagramını bir tabloya döktüğümüzde, giriş / çıkış ile önceki / sonraki durum ilişkilerine göre aşağıdaki durum tablosunu elde ederiz:



Şekil 2: Mealy tipi bir durum diyagramı.

Mevcut Durum	Sonraki Durum		Çıkış	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	g	f	0	1
g	a	f	0	1

Tablodan da e ve g durumlarının aynı satırlara sahip olduğunu görürüz. Yani aynı giriş değerleri için bu durumlar aynı sonraki durumu ve çıkış değerine sahiptirler. Dolayısıyla, bu durumlar eşdeğerdir ve bir tanesini kullanmak gereksizdir. Durum indirgemesi yapmak için sondaki g durumunu silebiliriz. Bu durumu tablodan çıkardıktan sonra ise e ve g durumları eşdeğer olduğundan tabloda g gördüğümüz yere e yazmamız gerekir:

Mevcut Durum	Sonraki Durum		Çıkış	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	e	f	0	1

Bu sadeleştirmeyi yaptıktan sonra ise, bu sefer d ile f durumlarının eşdeğer olduğunu görürüz. Tekrar indirgeme yapmak için f durumunu sileriz ve tabloda f gördüğümüz yere eşdeğeri olan d yazarız:

Mevcut Durum	Sonraki Durum		Çıkış	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	d	0	1
e	a	d	0	1

Elde ettiğimiz bu yeni durum tablosunda aynı olan başka satırlar yoktur. Dolayısıyla, bu tabloda daha fazla sadeleştirme yapamayız ve Şekil 2'deki sonlu durum makinesinin indirgenmiş hali yukarıdaki gibidir. Görmüş olduğunuz üzere durum diyagramının indirgenmiş hali 7 yerine 5 durum içermektedir.

- 1) İndirgenmiş durum diyagramını kendiniz çiziniz. İndirgenmiş durum diyagramı ile orijinal durum diyagramının birebir aynı sonuçları üreteceğine dikkat ediniz.
- 2) İndirgenmiş durum diyagramının da orijinal durum diyagramının da aynı sayıda flip flop ile gerçekleştirileceğine dikkat ediniz. İndirgenmiş durum diyagramını daha az sayıda flip flop ile gerçekleştirebilmek için durum sayısının 5'in altına düşmüş olması gerekirdi.

1.2 D FF ile Tasarım

D flip flopun karakteristik denkleminin $Q_{n+1} = D$ olmasından dolayı D flip flop ile tasarım son derece kolaydır. Durum tablosu ile önceki durum sonraki durum ilişkisini ortaya çıkardıktan sonra $Q_{n+1} = D$ ilişkisinden D girişine bağlanması gereken kombinasyonel devreyi hemen belirlemiş oluruz.

Örnek olarak Şekil 1'deki patern dedektörü uygulamasını D flip flop kullanarak gerçekleştirelim. Bunun için, öncelikle S_0, S_1, S_2, S_3 ile etiketlenmiş olan durumlara ikili kod vermemiz gerekir. Bunlar sırasıyla 00, 01, 10 ve 11 olsun. Toplam 4 tane durum olduğu için 2 adet D flip flop kullanmamız gerekir. Bu iki flip flopun çıkışları da sırasıyla A ve B olsun. Yani durumları $AB = 00, AB = 01, AB = 10$ ve $AB = 11$ çıkış değerleri ile temsil edelim. Buna göre, Şekil 1'deki durum diyagramını aşağıdaki gibi bir tabloya dönüştürebiliriz:

Önceki ders notunda analizde yapmış olduğumuz işlemin tam tersini yapıyoruz.

Şimdiki Durum		Giriş	Sonraki Durum		Çıkış
A	B	x	A	B	y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	0
1	1	0	0	0	1
1	1	1	1	1	1

Flip flopların girişleri sırasıyla D_A ve D_B olsun. D flip flopun karakteristik denkleminde $A_{n+1} = D_A$ ve $B_{n+1} = D_B$ olması gerektiğini biliyoruz. Yani yukarıdaki tabloda A ve B 'nin sonraki değerleri doğrudan flip flopların girişine bağlanması gereken kombinasyonel devreyi temsil eder. **Bu noktadan sonrası kombinasyonel devre tasarımıyla tamamen aynıdır.** Buna göre, flip flopların girişlerine bağlanması gereken kombinasyonel devreler aşağıdaki gibi ifade edilecektir:

$$D_A(A, B, x) = \sum(3, 5, 7) = \prod(0, 1, 2, 4, 6)$$

$$D_B(A, B, x) = \sum(1, 5, 7) = \prod(0, 2, 3, 4, 6)$$

Gördüğümüz üzere flip flop girişlerine bağlanacak kombinasyonel devreleri miniterimlerin toplamı veya maksiterimlerin çarpımı formunda gerçekleştirme ihtimalimiz vardır. Çarpımların toplamı formu daha az terim içerdiği için çarpımların toplamı sadeleştirmesini kullanarak kombinasyonel devreleri gerçekleştirelim. Sadeleştirmeyi önce D_A için yaparsak aşağıdaki Karnaugh haritasını elde ederiz:

		Bx			
		00	01	11	10
A	0			1	
	1		1	1	

Bu ise $D_A = Ax + Bx$ denkleminin karşılığı olacaktır. Aynı işlemi D_B için yaparsak, bu sefer de aşağıdaki Karnaugh haritasını elde ederiz:

	Bx			
	00	01	11	10
A				
0		1		
1		1	1	

Buradan da $D_B = Ax + B'x$ sonucunu elde ederiz.

Moore tipi bir sonlu durum makinesini gerçekleştirdiğimiz için çıkışın sadece A ve B durumlarına bağlı olması gerektiğini biliriz. Durum tablosuna baktığımızda ise y çıkışının sadece hem A hem de B flip flop çıkışı olduğunda 1 olduğunu görürüz. Buradan da y çıkış denkleminin $y = AB$ olması gerektiğini belirlememiz hiç de zor olmaz. Fakat, çıkışa ait kombinasyonel devreyi yukarıda yapmış olduğumuz gibi kombinasyonel devre tasarımıyla da belirleyebiliriz. Bunu yaparsak, çıkışa ait kombinasyonel devrenin miniterimlerin toplamı cinsinden aşağıdaki gibi ifade edilmesi gerektiğini görürüz:

$$y(A, B, x) = \sum(6, 7)$$

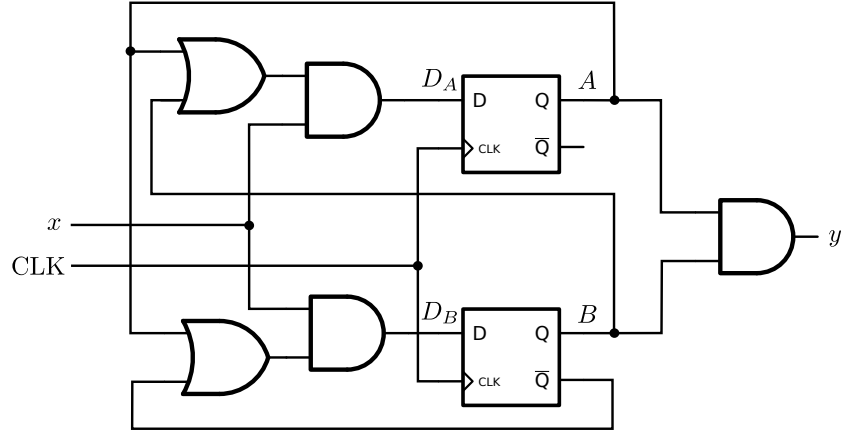
Bu devreyi en sade haliyle belirlemek için Karnaugh haritasını aşağıdaki gibi oluştururuz:

	Bx			
	00	01	11	10
A				
0				
1			1	1

Bu da bize yine $y = AB$ denklemini verir. Elde ettiğimiz sonuçları özetlersek, Şekil 1'deki patern dedektörüne ait flip flop girişleri ile çıkış Boole denklemleri çarpımların toplamı formunda sadeleştirilmiş halleriyle aşağıdaki gibi olacaktır:

$$\begin{aligned} D_A &= Ax + Bx \\ D_B &= Ax + B'x \\ y &= AB \end{aligned} \quad (1)$$

Bu denklemleri elde ettikten sonra devre şematini çıkarmak son derece kolaydır. Birinci flip flopun D_A girişine $Ax + Bx$ devresini bağlamamız gerekir. Dikkat ederkeniz $(A + B)x$ devresini bağlamak bir kapı daha az kullanmamızı sağlayacaktır.



Şekil 3: Moore tipi patern dedektörünün devre gerçekleştirimi.

Bu nedenle, birinci flip flopun girişine her iki flip flopun çıkışının VEYA kapısına sokulduktan sonra bu VEYA kapısının çıkışı ile x girişinin VE kapısına sokulduğu kombinyonel devreyi bağlamamız gerekir. Diğer flip flopun D_B girişine ise benzer şekilde $(A + B')x$ devresini bağlamamız gerekir. y çıkışı ise her iki flip flop çıkışının VE kapısına sokulmasıyla kolayca elde edilir. Buna göre, patern dedektörü devresinin şematığı Şekil 3'deki gibi olacaktır.

1.3 JK FF ile Tasarım

JK flip flop ile devre tasarımı D flip flopunki gibi doğrudan değildir. JK flip flop ile senkron devre tasarımı yapılırken durum tablosunda belirtilen sonraki durumu elde etmek için JK girişlerine ne bağlamamız gerektiğini belirlememiz gerekir. Bunun için genel olarak aşağıdaki tabloyu kullanalım:

Q_n	Q_{n+1}	J	K
0	0		
0	1		
1	0		
1	1		

Tablonun ilk satırında mevcut durum $Q_n = 0$ iken sonraki durumu da $Q_{n+1} = 0$ yapacak olan JK girişlerini belirlemek istiyoruz. Eğer $J = 0$, $K = 1$ girişi verirse flip flop resetlenecek ve sonraki durum istediğimiz gibi 0 olacaktır. Fakat, $J = 0$, $K = 0$ girişlerini verirse bu sefer de önceki durum tutulacağından sonraki durum yine 0 olacaktır. Dolayısıyla, $J = 0$ olduğu müddetçe K değerinden bağımsız olarak ilk satırdaki şartı sağlamış oluruz. Yani, ilk satırda J yerine 0; K yerine fark etmez anlamında X koymamız gerekir.

İkinci satırda ise mevcut durum 0 iken sonraki durumu 1 yapmak istiyoruz. Bunu yine iki şekilde yapabiliriz: i) $J = 1, K = 0$ ile flip flop set ederiz; ii) $J = 1, K = 1$ ile önceki durumun tersini alırız. Görüldüğü üzere bu sefer $J = 1$ olduğu müddetçe K değeri önemsizdir. Dolayısıyla, ikinci satırda J yerine 1; K yerine X koymamız gerekir.

Üçüncü satırda mevcut durum 1 iken sonraki durumu 0 yapmak istiyoruz. Bunu yine iki şekilde yapabiliriz: i) $J = 0, K = 1$ ile flip flop reset ederiz; ii) $J = 1, K = 1$ ile önceki durumun tersini alırız. Bu sefer de $K = 1$ olduğu müddetçe J önemsizdir. Dolayısıyla, üçüncü satırda J yerine X ; K yerine 1 koymamız gerekir.

Son satırda ise mevcut durum 1 iken sonraki durumu da 1 yapmamız istenmektedir. Yine iki alternatifimiz var: i) $J = 1, K = 0$ ile flip flop set etmek; ii) $J = 0, K = 0$ ile önceki durumu korumak. Buna göre, $K = 0$ olduğu müddetçe J önemsizdir. Dolayısıyla, son satırda J yerine X ; K yerine 0 koymamız gerekir.

Bu değerleri yerine koyduğumuzda Tablo 1'i elde ederiz. Tablo 1 aynı zamanda JK flip flopun **uyarım tablosu**² olarak da adlandırılır. İşte JK flip flop ile tasarım yaparken durum tablosunda J ve K girişlerinin değerlerinin ne olması gerektiği bu uyarım tablosuna bakılarak belirlenir. Bu nedenle, JK flip flop ile tasarım yaparken uyarım tablosunu da bir köşeye not etmeniz durum tablosunu doldururken hata yapmanıza engel olacaktır.

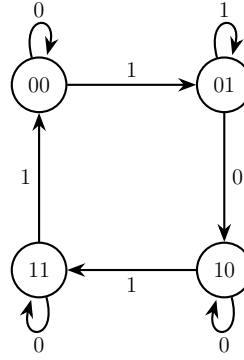
Q_n	Q_{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

Tablo 1: JK flip flopun uyarım tablosu.

Yukarıda görmüş olduğumuz üzere JK flip flop ile tasarım yaparken $Q_n = 0$ ve $Q_{n+1} = 0$ için durum tablosunda $J = 0$ ve $K = X$ almamız gerekir. Fakat, $J = 0, K = 1$ veya $J = 0, K = 0$ alırsanız tasarladığınız devre yine doğru çalışacaktır. Ancak bu durumda $K = X$ almayarak devre sadeleştirmesinde kullanılabilecek ihtimalleri azalttığımız için devreyi en sade haliyle kurmamış olacaksınız! (Fark etmez durumlarının Karnaugh haritası ile sadeleştirmede ne kadar faydalı olduğunu daha önce görmüştük).

Bu uyarım tablosundaki diğer satırlar için de benzerdir. Tablo 1'deki uyarım tablosu içerdiği fark etmez (X) durumlarıyla devreyi en sade haliyle kurmanızı sağlar.

²ing. *excitation table*



Şekil 4: JK flip flop kullanılarak gerçekleştirilecek olan durum diyagramı.

Örnek olarak Şekil 4’de verilen durum diyagramını JK flip flop kullanarak gerçekleştirelim. Durum diyagramından görüldüğü üzere devrenin bir girişi varken çıkışı yoktur. Girişi x ile gösterelim. Elimizde 4 durum olduğu için 2 tane JK flip flop kullanmamız gerekecektir. Bu flip flopların çıkışları sırasıyla A ve B olsun. Flip flopların girişlerini de J_A , K_A ve J_B , K_B ile etiketleyelim. Durum diyagramına bakarak devreye ait önceki / sonraki durum ilişkisi aşağıdaki gibi elde edilir:

Şimdiki Durum		Giriş x	Sonraki Durum		Flip Flop Girişleri			
A	B		A	B	J_A	K_A	J_B	K_B
0	0	0	0	0				
0	0	1	0	1				
0	1	0	1	0				
0	1	1	0	1				
1	0	0	1	0				
1	0	1	1	1				
1	1	0	1	1				
1	1	1	0	0				

Şimdi yapmamız gereken şimdiki durumdan sonraki durumlara geçişlere bakarak flip flop girişlerinin ne olması gerektiğini belirlemektir. Bunun için Tablo 1’de verilen uyarım tablosundan faydalanırız. Mesela ilk satırda $A_n = 0$ iken $A_{n+1} = 0$ olmaktadır. Uyarım tablosundan bu durumda $J_A = 0$ ve $K_A = X$ almamız gerektiğini görürüz. İlk satırda B de aynı ilişkiye sahip olduğundan benzer şekilde $J_B = 0$ ve $K_B = X$ olmalı.

İkinci satırda A yine yukarıdaki gibi olduğundan tabloya tekrar $J_A = 0$ ve $K_A = X$ yazmamız gerekir. B içinse $B_n = 0$ iken $B_{n+1} = 1$ olduğundan uyarım tablosuna bakarak $J_B = 1$ ve $K_B = X$ almamız gerektiğini görürüz. Aynı işlemi benzer şekilde geri kalan tüm satırlar için tekrarladığımızda gerçekleştirmek istediğimiz devreye ait durum tablosunu aşağıdaki gibi elde ederiz:

Şimdiki Durum		Giriş x	Sonraki Durum		Flip Flop Girişleri			
A	B		A	B	J_A	K_A	J_B	K_B
0	0	0	0	0	X	0	X	
0	0	1	0	1	0	X	1	X
0	1	0	1	0	1	X	X	1
0	1	1	0	1	0	X	X	0
1	0	0	1	0	X	0	0	X
1	0	1	1	1	X	0	1	X
1	1	0	1	1	X	0	X	0
1	1	1	0	0	X	1	X	1

Şimdi flip flop girişlerine bağlanacak olan kombinyonel devreleri en sade halleriyle belirlememiz gerekir. **Bu noktadan sonra yapılacaklar kombinyonel devre tasarımıyla tamamen aynıdır.** Flip flop girişlerine bağlanacak devreleri miniterimlerin toplamı cinsinden ifade edersek aşağıdaki eşitlikleri elde ederiz:

$$J_A(A, B, x) = \sum(2) + d(4, 5, 6, 7)$$

$$K_A(A, B, x) = \sum(7) + d(0, 1, 2, 3)$$

$$J_B(A, B, x) = \sum(1, 5) + d(2, 3, 6, 7)$$

$$K_B(A, B, x) = \sum(2, 7) + d(0, 1, 4, 5)$$

Burada $d(\cdot)$ ile ifade edilen değerler önceki notlardan hatırlayacağınız üzere fark etmez durumlarını temsil etmektedir. Buna göre J_A için Karnaugh haritasını aşağıdaki gibi oluştururuz:

		Bx			
		00	01	11	10
A	0				1
	1	X	X	X	X

Tabloyu kullanarak gerekli sadeleştirmeleri yaptığımızda $J_A = Bx'$ eşitliğini elde ederiz.

K_A için Karnaugh haritası aşağıdaki gibi olacaktır:

	Bx	00	01	11	10
A	0	X	X	X	X
	1			1	

Buradan da $K_A = Bx$ eşitliğini elde ederiz.

J_B içinse Karnaugh haritası aşağıdaki gibi olacaktır:

	Bx	00	01	11	10
A	0		1	X	X
	1		1	X	X

Buradan da $J_B = x$ eşitliğini elde ederiz.

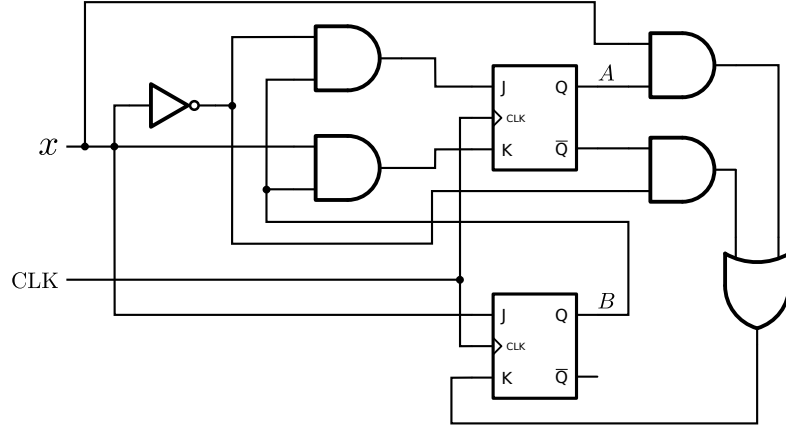
K_B için Karnaugh haritası aşağıdaki gibi olacaktır:

	Bx	00	01	11	10
A	0	X	X		1
	1	X	X	1	

Buradan da $K_B = Ax + A'x'$ eşitliğini elde ederiz. Elde etmiş olduğumuz sadeleştirilmiş Boole eşitliklerini aşağıdaki gibi sıralayabiliriz:

$$\begin{aligned}
 J_A &= Bx' \\
 K_A &= Bx \\
 J_B &= x \\
 K_B &= Ax + A'x'
 \end{aligned} \tag{2}$$

Eşitlik 2 kullanılarak oluşturulan devre şematığı Şekil 5'de görüldüğü gibidir. (NOT: $(A \oplus x)' = (Ax' + A'x)' = (A' + x)(A + x') = Ax + A'x'$ olduğu için aynı devreyi $K_B = (A \oplus x)'$ eşitliğini kullanarak standart olmayan formda da kurabilirsiniz.)



Şekil 5: JK flip flop kullanılarak oluşturulan devrenin şematığı.

Q_n	Q_{n+1}	T
0	0	0
0	1	1
1	0	1
1	1	0

Tablo 2: T flip flopun uyarım tablosu.

1.4 T FF ile Tasarım

T flip flop ile tasarım yaparken de tıpkı JK flip flopta olduğu gibi flip flopun T girişinin ne olması gerektiğini belirleyecek bir uyarım tablosu oluşturmamız lazım. Hatırlayacağınız üzere T flip flop $T = 0$ için önceki durumu korurken, $T = 1$ için önceki durumu terslendirmekteydi. Çalışma prensibi basit olduğu için aşağıdaki uyarım tablosunu doldurmak da son derece basit olacaktır:

Q_n	Q_{n+1}	T
0	0	
0	1	
1	0	
1	1	

İlk ve son satırlarda mevcut durum korunmaktadır. Bu satırlar için bunu sağlayacak T girişi 0 olmalıdır. İkinci ve üçüncü satırlarda ise mevcut durum terslendirilmektedir. Bu nedenle bu satırlarda $T = 1$ olmalıdır. Bunları yerine koyduğumuzda T flip flopun uyarım tablosunu aşağıdaki gibi elde ederiz:

Durum tablosunda T giriş değerlerini doldururken bu tabloya bakmanıza gerek bile yok. Eğer durum değişikliği varsa T terine 1; eğer durum aynı kalıyorsa T yerine 0 yazmanız yeterlidir.

Flip flopların önemli uygulamalarından biri de sayıcılardır. Bu konuyu haftaya detaylı olarak işleyeceğiz, fakat şimdi örnek olarak T flip flop kullanarak 0'dan 7'ye kadar sayan bir sayıcı tasarlayalım. Tabii ki bu sayma işi ikili tabanda olmak zorundadır. 0 ile 7 arasındaki sayıları 3-bitle ifade edebileceğimizden sayma işi $000 \rightarrow 001 \rightarrow \dots \rightarrow 111$ şeklinde olmalı. Burada her bir sayı bir durumu ifade etmeli. Yani toplamda 8 durumumuz olmalı ve en son 111 durumundan sonra sayıcı tekrar başa dönmeli. Bu problem için durum diyagramı oluşturmamıza gerek bile yok. Önceki / sonraki durum ilişkisi $000 \rightarrow 001 \rightarrow 010 \rightarrow 011 \rightarrow \dots \rightarrow 111 \rightarrow 000 \rightarrow \dots$ şeklinde son derece basit. Her bir flip flop çıkışını A, B, C ile ifade edelim ve A flip flopu MSB'yi temsil etsin. Bu flip flopların girişleri de sırasıyla T_A, T_B ve T_C olsun. Buna göre önceki / sonraki durum ilişkisini durum tablosunda aşağıdaki gibi gösterebiliriz:

Şimdiki Durum			Sonraki Durum			Flip Flop Girişleri		
A	B	C	A	B	C	T_A	T_B	T_C
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			

Önceki / sonraki durum ilişkisini belirledikten sonra flip flop girişlerini doldurabiliriz. Yapmamız gereken son derece basit. Birinci satırda A, B, C durumlarına baktığımızda sadece C durumunun değişiklik gösterdiğini görürüz. Buna göre, bu satırda $T_A = T_B = 0$ iken $T_C = 1$ olmalı. İkinci satıra baktığımızda hem B hem de C durumlarının değiştiğini görürüz. Bu nedenle, bu satırda $T_A = 0$ iken $T_B = T_C = 1$ olmalı. Benzer şekilde diğer satırları da doldurduğumuzda aşağıdaki durum tablosunu elde ederiz:

Şimdiki Durum			Sonraki Durum			Flip Flop Girişleri		
A	B	C	A	B	C	T_A	T_B	T_C
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1

Bu noktadan sonra yapılacaklar yine kombinyonel devre tasarımı olacaktır. Bir kez daha miniterimleri kullanırsak, flip flop girişlerine bağlanması gereken kombinyonel devre eşitliklerini aşağıdaki gibi ifade edebiliriz:

$$T_A(A, B, C) = \sum(3, 7)$$

$$T_B(A, B, C) = \sum(1, 3, 5, 7)$$

$$T_C(A, B, C) = 1$$

T_C sabit olduğu için sadeleştirmeye gerek yoktur. Fakat, T_A ve T_B Boole eşitlikleri sadeleştirilmelidir. T_A için yapılacak sadeleştirmeye ait Karnaugh haritası aşağıdaki gibi olmalıdır:

	BC			
	00	01	11	10
A				
0			1	
1			1	

Buradan $T_A = BC$ sadeleştirilmiş eşitliğini elde ederiz.

T_B için ise Karnaugh haritası aşağıdaki gibi olacaktır:

	BC			
	00	01	11	10
A				
0		1	1	
1		1	1	

Buradan da $T_B = C$ eşitliğini elde ederiz. Sonuç olarak 3-bitlik sayıcının T flip flop ile gerçekleştirimi için gerekli bağlantılar aşağıdaki gibi olacaktır:

$$\begin{aligned} T_A &= BC \\ T_B &= C \\ T_C &= 1 \end{aligned} \tag{3}$$

Bu devrenin şematığının oluşturulması son derece kolaydır. *Devrenin şematığını kendiniz oluşturunuz. Sonrasında da flip flop çıkışlarına led bağlayarak online veya tercih ettiğiniz bir dijital simülasyon yardımıyla devrenin simülasyonunu yapınız. Ledlerin doğru sırayla yanıp yanmadığını kontrol ediniz.*

2 FF Karakteristik Denklemlerinin Elde Edilmesi

Daha önce JK flip flopu D flip flopun girişine $D = JQ' + K'Q$ devresini bağlayarak elde edebileceğimizi görmüştük. D flip flop için $Q_{n+1} = D$ ilişkisi olduğundan, JK flip flopun karakteristik denklemini de

$$Q_{n+1} = JQ' + K'Q \tag{4}$$

olarak bulmuştuk. Bu denklemin nereden geldiğini belirlemek için flip flop ile tasarım yapmayı bilmemiz gerekiyordu. Şimdi tasarım yapmayı gördüğümüze göre bu denklemi elde edebiliriz.

Bunun için D flip flop kullanarak JK flip flop elde etmek için tasarım yapalım. Tasarımımızın girişleri J ve K olacaktır. Bu J ve K girişleri ile D flip flopun Q çıkışını bir kombinasyonel devre ile bağladığımızda devre tam olarak JK flip flopun doğruluk tablosunda olduğu gibi çalışmalıdır. Yani $J = K = 0$ için Q değeri sonraki durumda korunurken, $J = K = 1$ için sonraki durum Q' olmalı. Ayrıca, $J = 1$, $K = 0$ için sonraki durum mevcut durumdan bağımsız olarak 1 olmalıyken, $J = 0$, $K = 1$ için sonraki durum 0 olmalı. Bulmak istediğimiz bu şartları sağlayan bir $D(J, K, Q)$ kombinasyonel devresi. Bu devreyi bulmak için J , K girişleri ile Q mevcut durumunun olası tüm kombinasyonlarını içeren durum tablosu aşağıdaki gibi doldurulur:

Girişler		Şimdiki Durum	Sonraki Durum
J	K	Q	$Q_{n+1} = D$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Bu durum tablosundan D girişine bağlanması gereken kombinasyonel devreyi miniterimler cinsinden aşağıdaki gibi elde ederiz:

$$D(J, K, Q) = \sum(1, 4, 5, 6)$$

Gerekli sadeleştirmeleri aşağıdaki Karnaugh haritası yardımıyla yaparız:

		KQ			
		00	01	11	10
J	0		1		
	1	1	1		1

Buradan da

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

Boole eşitliğini elde ederiz.

Şimdi ise benzer şekilde T flip flopun karakteristik denklemini elde edelim. Hatırlayacağımız üzere T flip flopunu D flip flop kullanarak oluşturmak istediğimizde D flip flopun girişine $D = T \oplus Q$ kombinasyonel devresini bağlamamız gerektiğini görmüştük. Ayrıca, yine $Q_{n+1} = D$ ilişkisini kullanarak T flip flopa ait karakteristik denklemini

$$Q_{n+1} = T \oplus Q \quad (5)$$

olarak elde etmiştik.

D flip flop ile tasarım yaparak aynı denklemi elde etmek için tasarlayacağımız devrenin T girişi olmalı. Bu T girişi tıpkı T flip flopun T girişi gibi çalışmalı: $T = 0$ için sonraki durum yine Q olurken, $T = 1$ için sonraki durum Q' olmalı. Bu ilişkiyi kullanarak durum tablosu aşağıdaki gibi doldurulur:

Giriş	Şimdiki Durum	Sonraki Durum
T	Q	$Q_{n+1} = D$
0	0	0
0	1	1
1	0	1
1	1	0

Bu durum tablosu

$$D(T, Q) = T \oplus Q$$

işlemine karşılık gelmektedir (XOR işleminin doğruluk tablosu tam olarak yukarıdaki gibidir. İsterseniz 2×2 Karnaugh haritası oluşturarak sonucu teyit edebilirsiniz).

Versiyon Notları

ver. 1.1:

- Sayfa 11 ve 12'deki tablolarda sonraki durumlar D flip flop örneğindeki sonraki durumlarla aynıydı. Bu düzeltildi. Bunu bildirdikleri için Levent Elbaş ve Cansu Beken'e teşekkür ederim.
- Bazı tablolardaki boşluklar ve dikey çizgiler düzeltildi.