

Mantıksal Sistem Tasarımı – BLM 201

Hafta 6: Sonlu Durum Makinaları



Fenerbahçe Üniversitesi

Öğretim Elemanları

Öğretim Üyesi: Dr. Vecdi Emre Levent

Ofis: 311

Email: emre.levent@fbu.edu.tr

Asistan: Arş. Gör. Uğur Özbalkan

Ofis: 311

Email: ugur.ozbalkan@fbu.edu.tr

Ders Planı

- Sonlu Durum Makinaları (Finite State Machine)

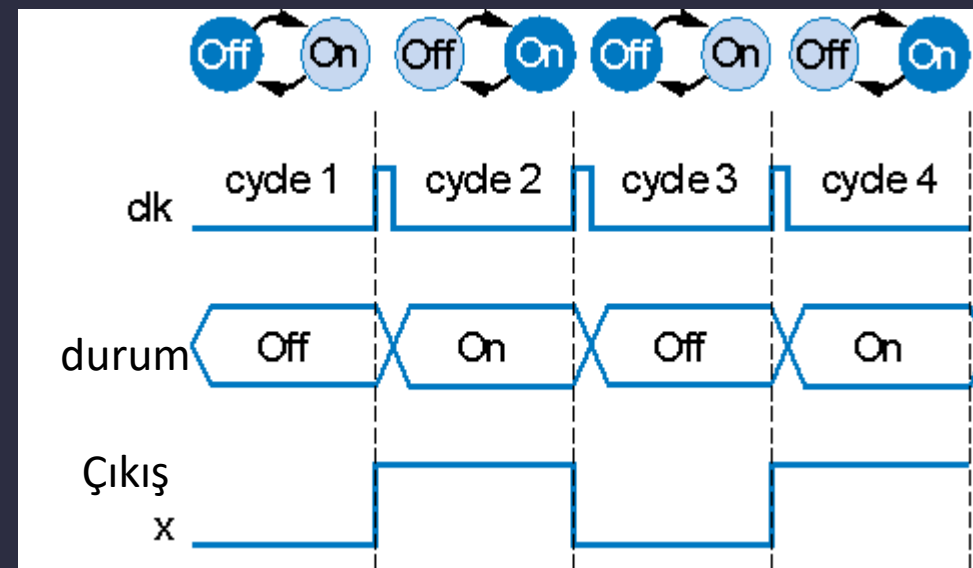
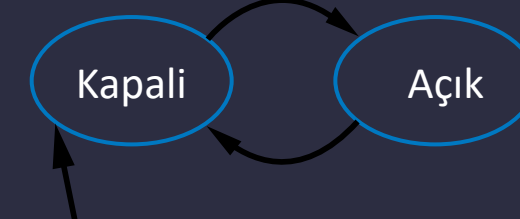
Ardışık Devrenin Davranışını Belirleme FSM

- Finite-State Machine (FSM)
 - Ardışık devrenin durumlara göre davranışının gösterim biçimidir
 - Durumları ve durumlar arası geçişler listelenir
 - Örnek: X isminde bir çıkış sinyali olsun ve her clock cycle'da değeri değişsin
 - İki durum: "Off" ($x=0$), ve "On" ($x=1$)
 - Off durumundan On durumuna ve On durumundan Off durumuna geçişler bulunmaktadır.
 - Başlangıç durumu bir ok ile belirtilir.

Çıkış: x

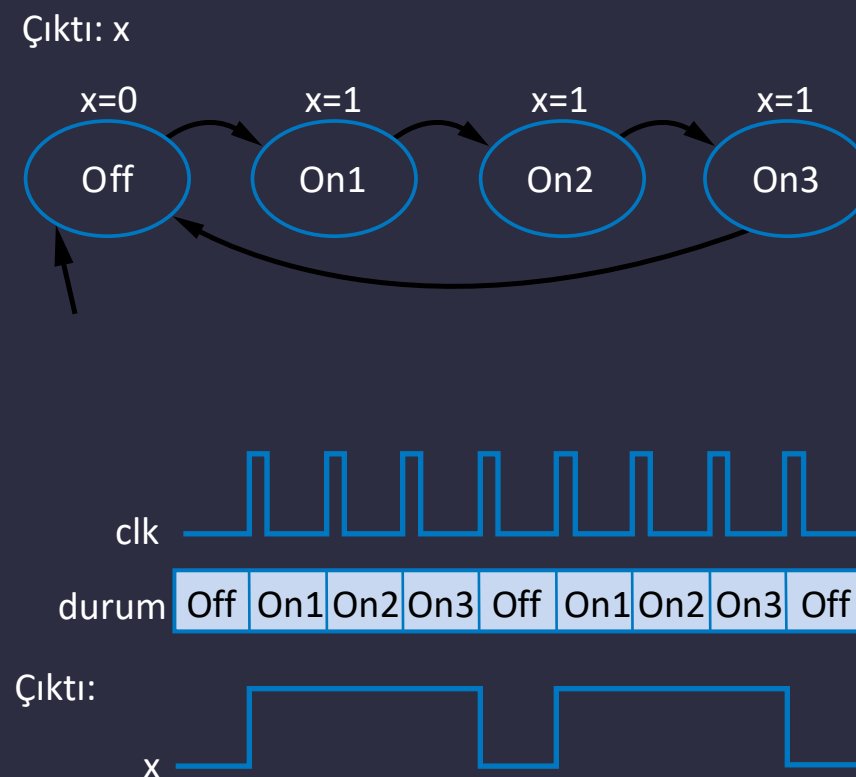
x=0

x=1



FSM Örneği: 0,1,1,1, tekrarı

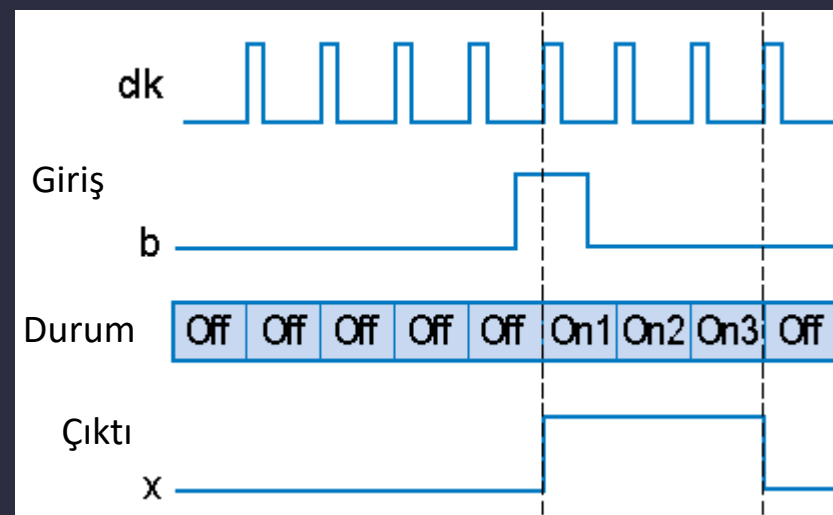
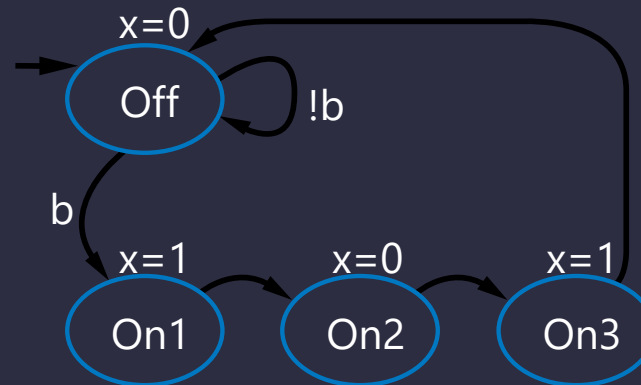
- Sırayla 0, 1, 1, 1, 0, 1, 1, 1, ... çıktılarını veren bir devre tasarlanacaktır.
 - Her bir değer bir clock cycle'da çıkmaktadır.
- FSM ile tasarlanabilir
 - 4 durum bulunur



FSM Örneği

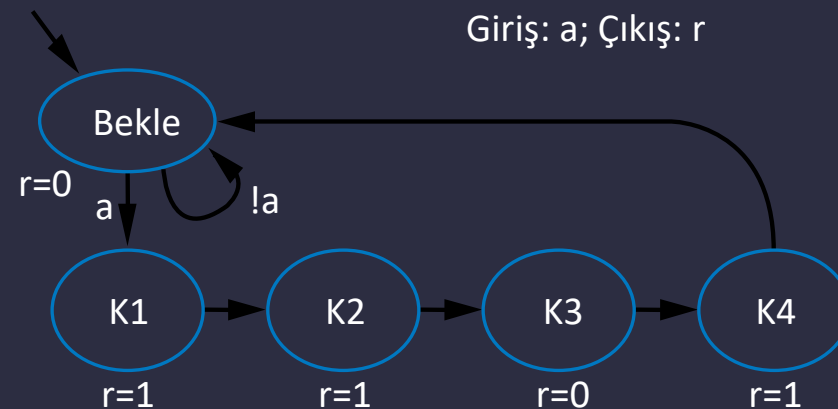
b isminde bir bitlik giriş alıp, b girişi aktif olduğunda 101 pattern'ini çıktı verip, tekrar b girişini bekleyen bir durum makinası tasarımı yapınız.

Girişler: b; Çıktılar: x



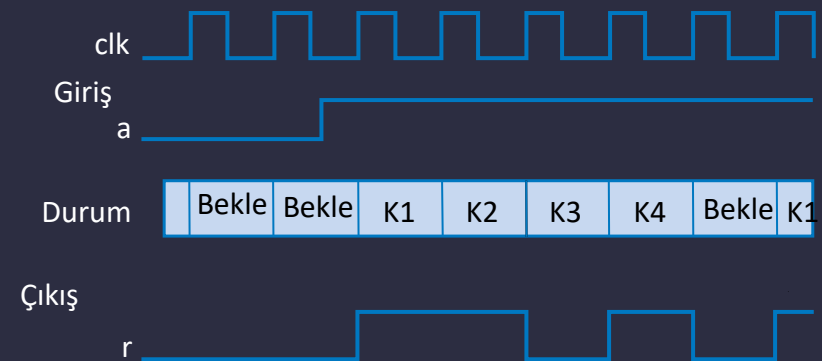
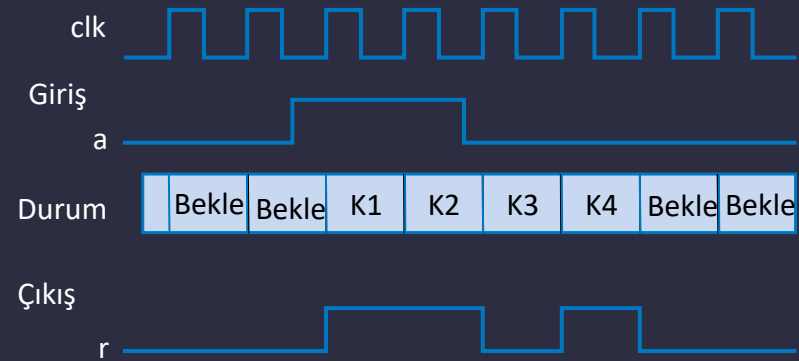
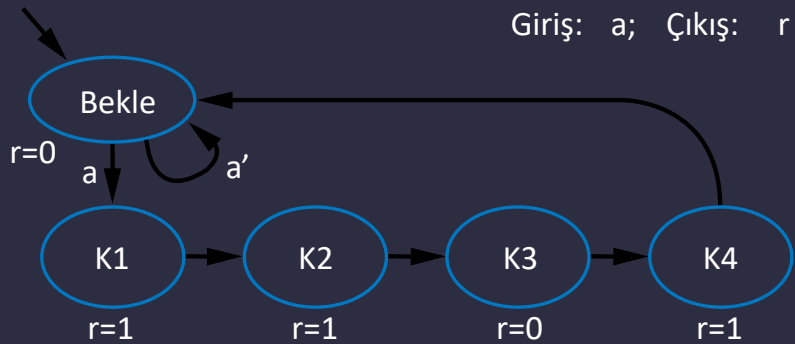
FSM Örneği: Araba Anahtarı

- Araba anahtarlarının içerisinde anahtarın kimliğini tutan çip bulunur.
 - Arabaya anahtar takıldığında, anahtardan kimlik bilgisini ister.
 - Anahtar kimlik bilgisini gönderir, doğru değilse araç çalışmaz
- FSM
 - İstek yapılmasını bekle ($a=1$)
 - Kimlik'i gönder (Örneğin, 1101)



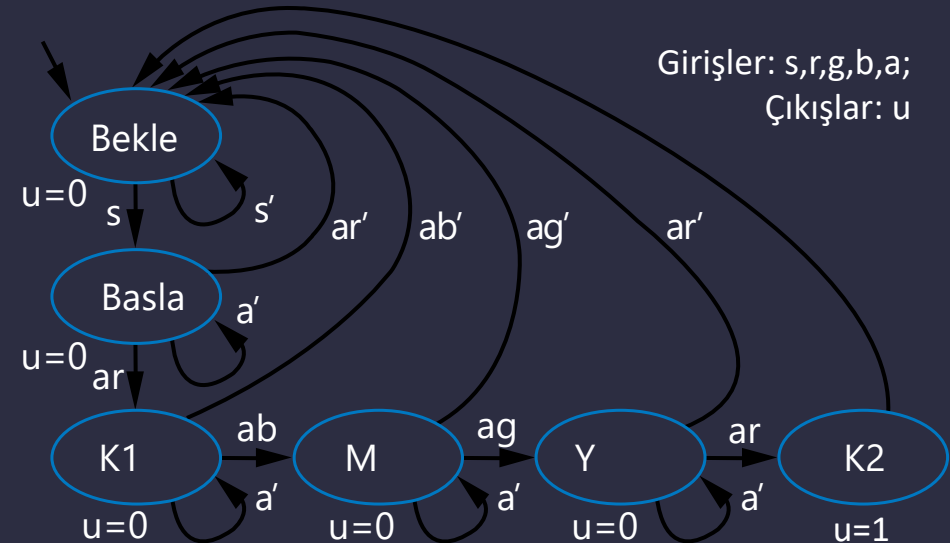
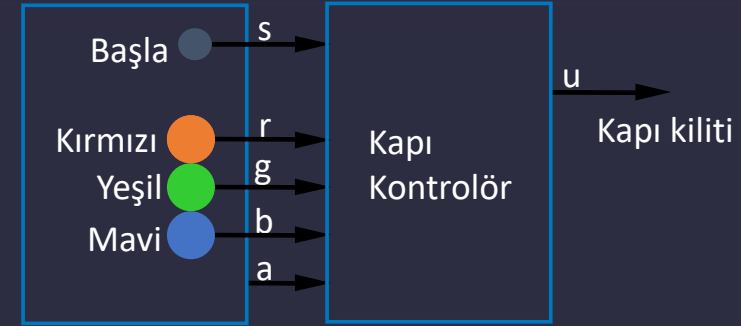
FSM Örneği: Araba Anahtarı

- FSM Zamanlama'ları

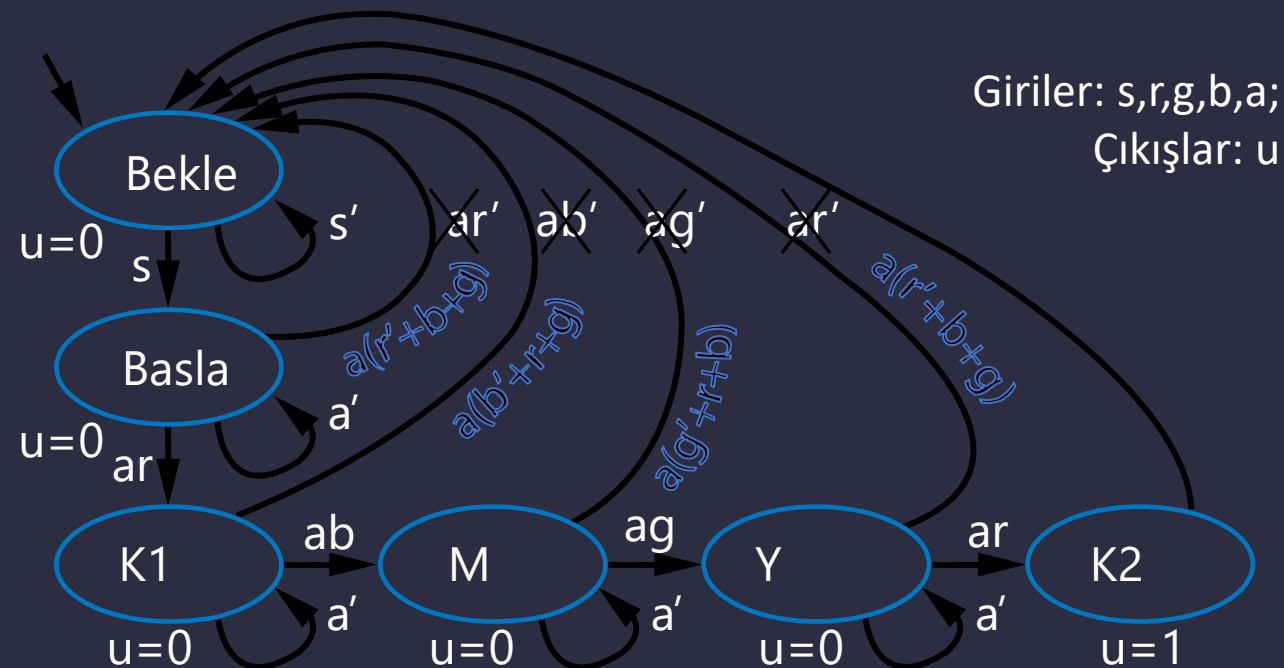


FSM Örneği: Kod Tespit Edici

- Bir kapı sadece aşağıdaki buton sıralamasına basıldığı zaman açılmaktadır ($u=1$)
 - başla, kırmızı, mavi, yeşil, kırmızı
- s, r, g, b
 - a girişi ise, diğer renkteki butonlar
- FSM
 - Bekle durumunda iken, başlat geldiğinde başla
 - Başlata basıldığında
 - Eğer kırmızı ise, K1'e
 - Sonra, mavi ise, M
 - Sonra, yeşil ise, Y
 - Sonra, kırmızı ise, K2
 - Bu durumda kapı açılacak sinyal üretilir ($u=1$)
 - Herhangi yanlış bir girişte bekle durumuna geri dönülür



FSM Örneği: Kod Tespit Edici, İyileştirmeler

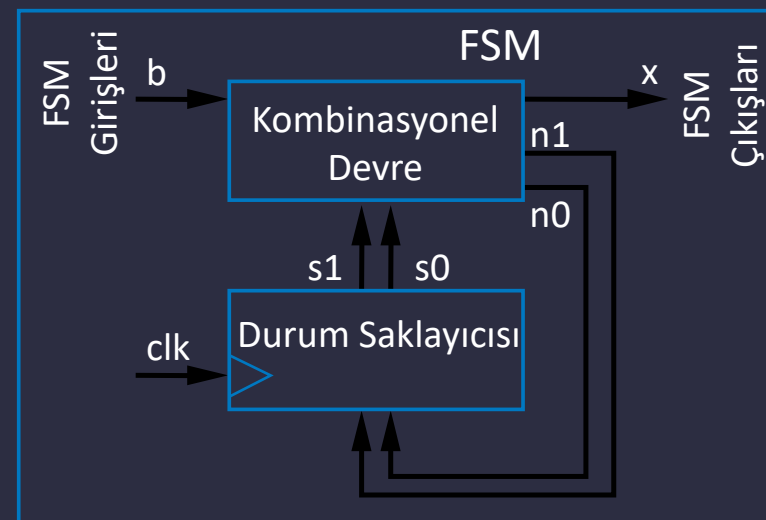
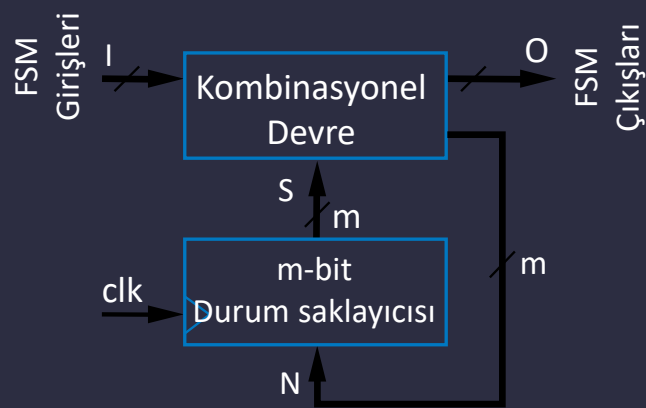
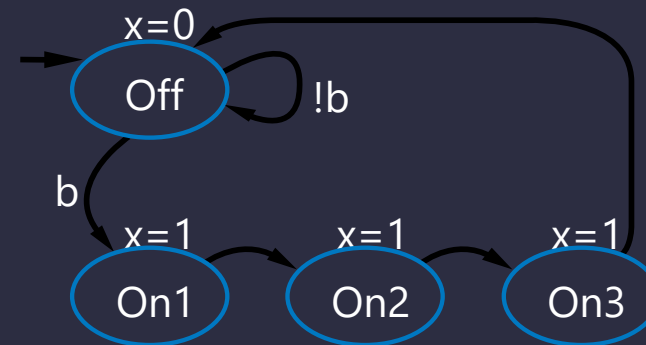


Yeni durum makinasında, yanlış butona basıldığında, "Bekle" durumuna geri dönlmektedir.

FSM Mimarisi

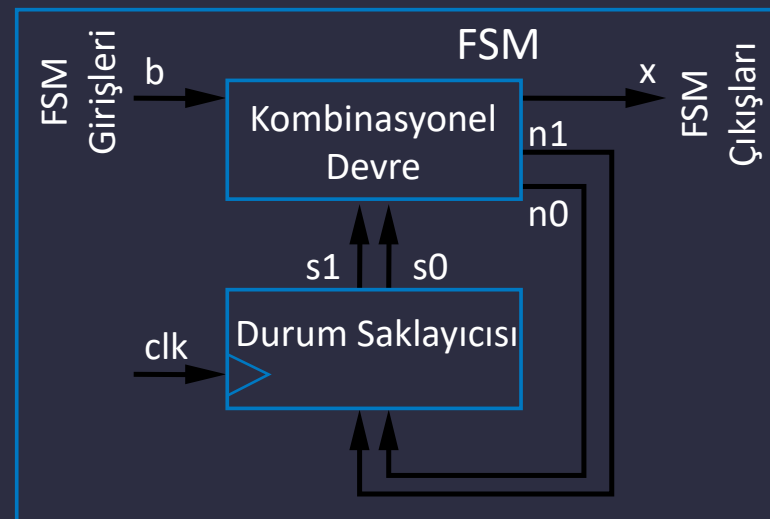
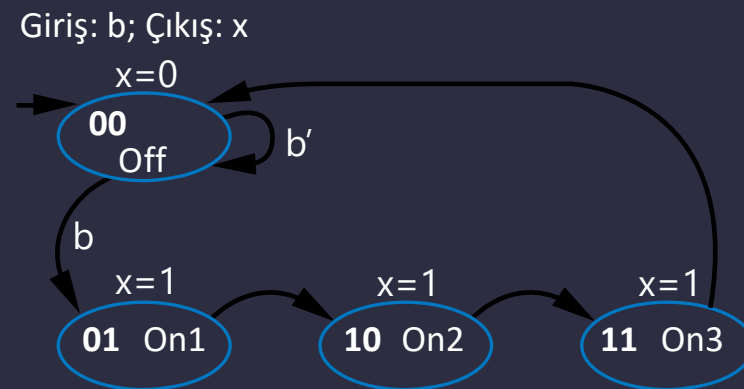
- FSM ardışık devre olarak nasıl gerçekleştirilir?
 - Durum saklayıcısı - şu anki durumu tutmak için
 - Kombinasyonel devre – çıkışı ve sonraki durumu hesaplamak için

Giriş: b; Çıkış: x



FSM Tasarım Örneği

- Adım 1: Gereksinimleri analiz
- Adım 2: Mimari çıkart
 - 2-bit durum saklayıcısı (4 durum için)
 - Giriş b, çıkış x
 - Sonraki durum sinyalleri n1, n0
- Adım 3: State'leri kodla

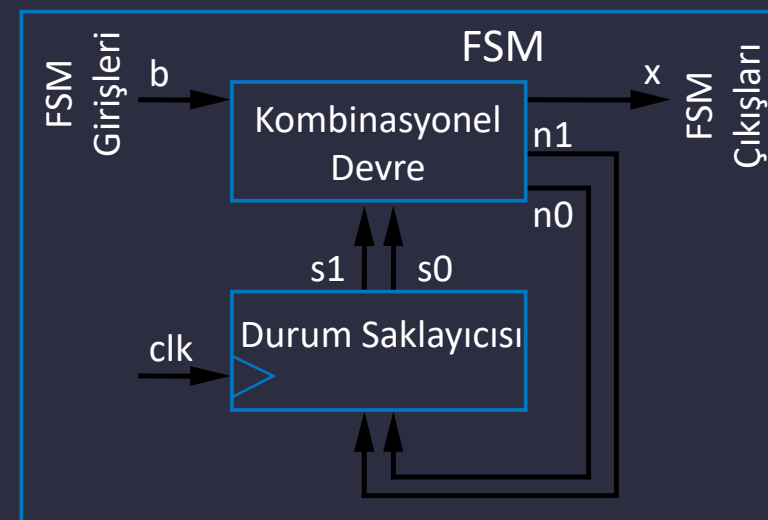
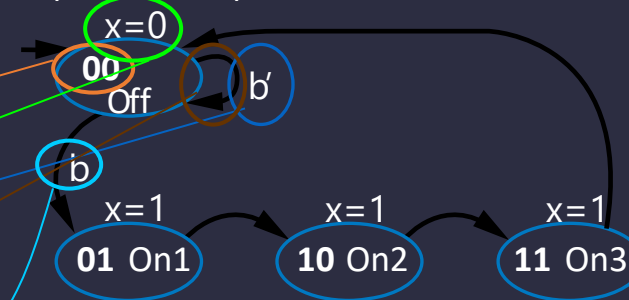


FSM Tasarım Örneği

- Adım 4: Doğruluk Tablosu oluştur

	Girişler			Çıktılar		
	s1	s0	b	x	n1	n0
<i>Off</i>	0	0	0	0	0	0
	0	0	1	0	0	1
<i>On1</i>	0	1	0	1	1	0
	0	1	1	1	1	0
<i>On2</i>	1	0	0	1	1	1
	1	0	1	1	1	1
<i>On3</i>	1	1	0	1	0	0
	1	1	1	1	0	0

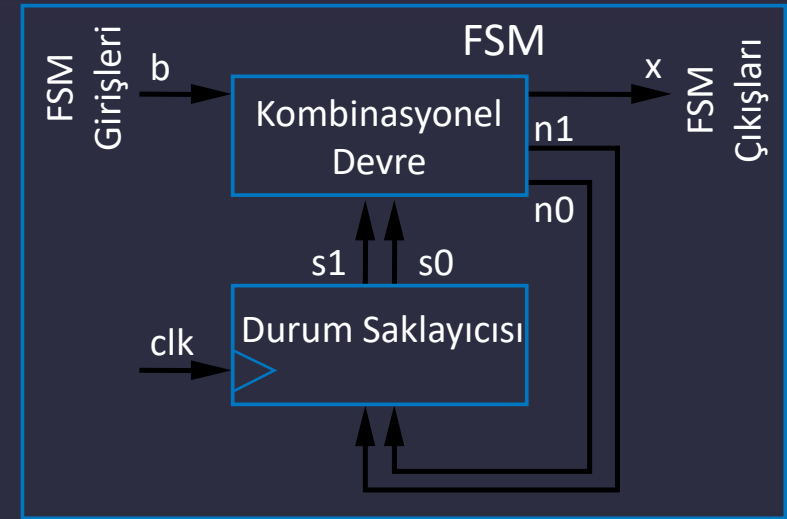
Inputs: b; Outputs: x



FSM Tasarım Örneği

- Adım 5: Kombinasyonel Devre Gerçeklemesi

	Girişler			Çıkışlar		
	s1	s0	b	x	n1	n0
<i>Off</i>	0	0	0	0	0	0
	0	0	1	0	0	1
<i>On1</i>	0	1	0	1	1	0
	0	1	1	1	1	0
<i>On2</i>	1	0	0	1	1	1
	1	0	1	1	1	1
<i>On3</i>	1	1	0	1	0	0
	1	1	1	1	0	0



$$x = s1 \mid s0$$

$$n1 = s1's0b' \mid s1's0b \mid s1s0'b' \mid s1s0'b$$

$$n1 = s1's0 \mid s1s0'$$

$$n0 = s1's0'b \mid s1s0'b' \mid s1s0'b$$

$$n0 = s1's0'b \mid s1s0'$$

FSM Tasarım Örneği

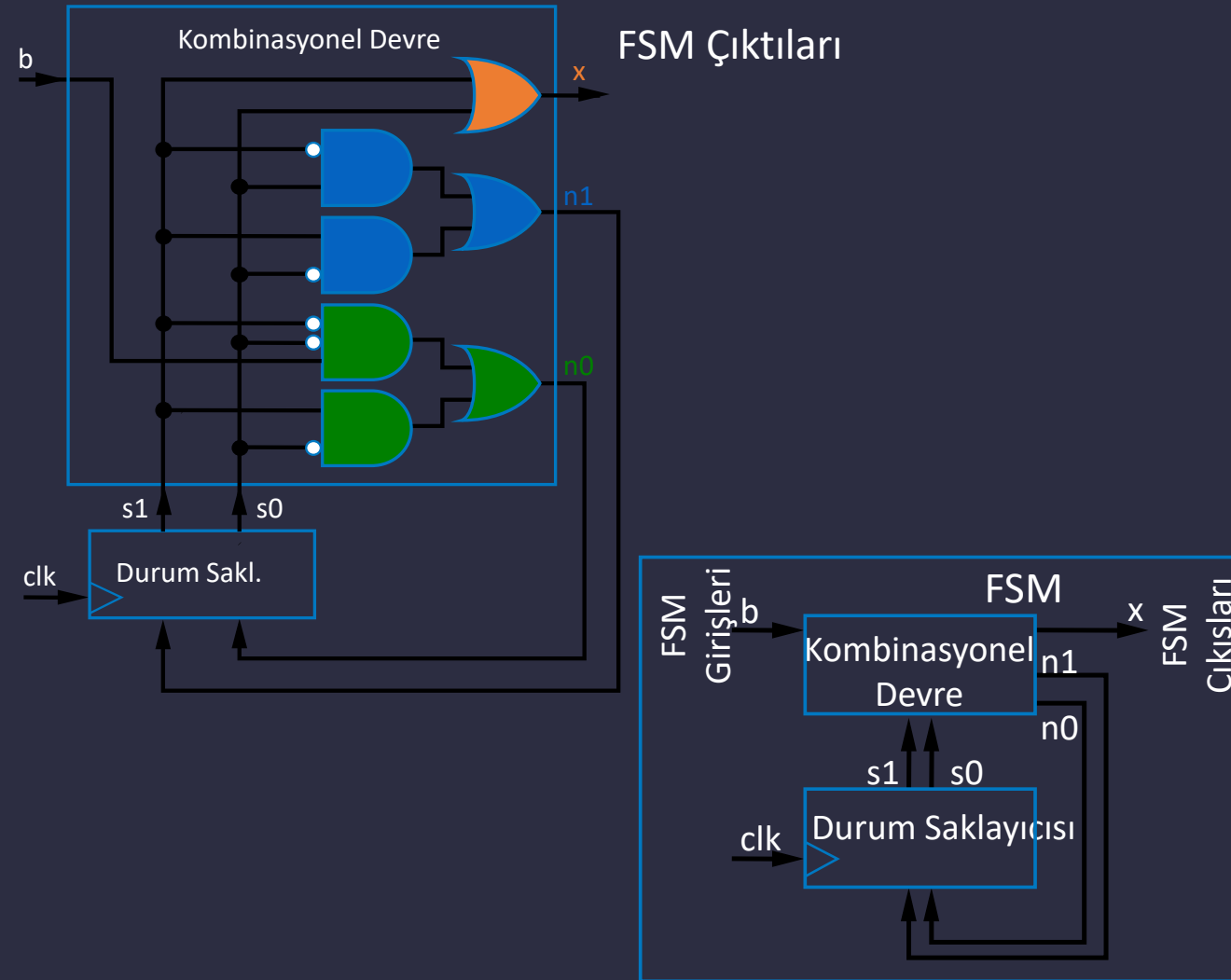
- Adım 5: Kombinasyonel devreyi oluştur

	Girişler			Çıktılar		
	s1	s0	b	x	n1	n0
<i>Off</i>	0	0	0	0	0	0
	0	0	1	0	0	1
<i>On1</i>	0	1	0	1	1	0
	0	1	1	1	1	0
<i>On2</i>	1	0	0	1	1	1
	1	0	1	1	1	1
<i>On3</i>	1	1	0	1	0	0
	1	1	1	1	0	0

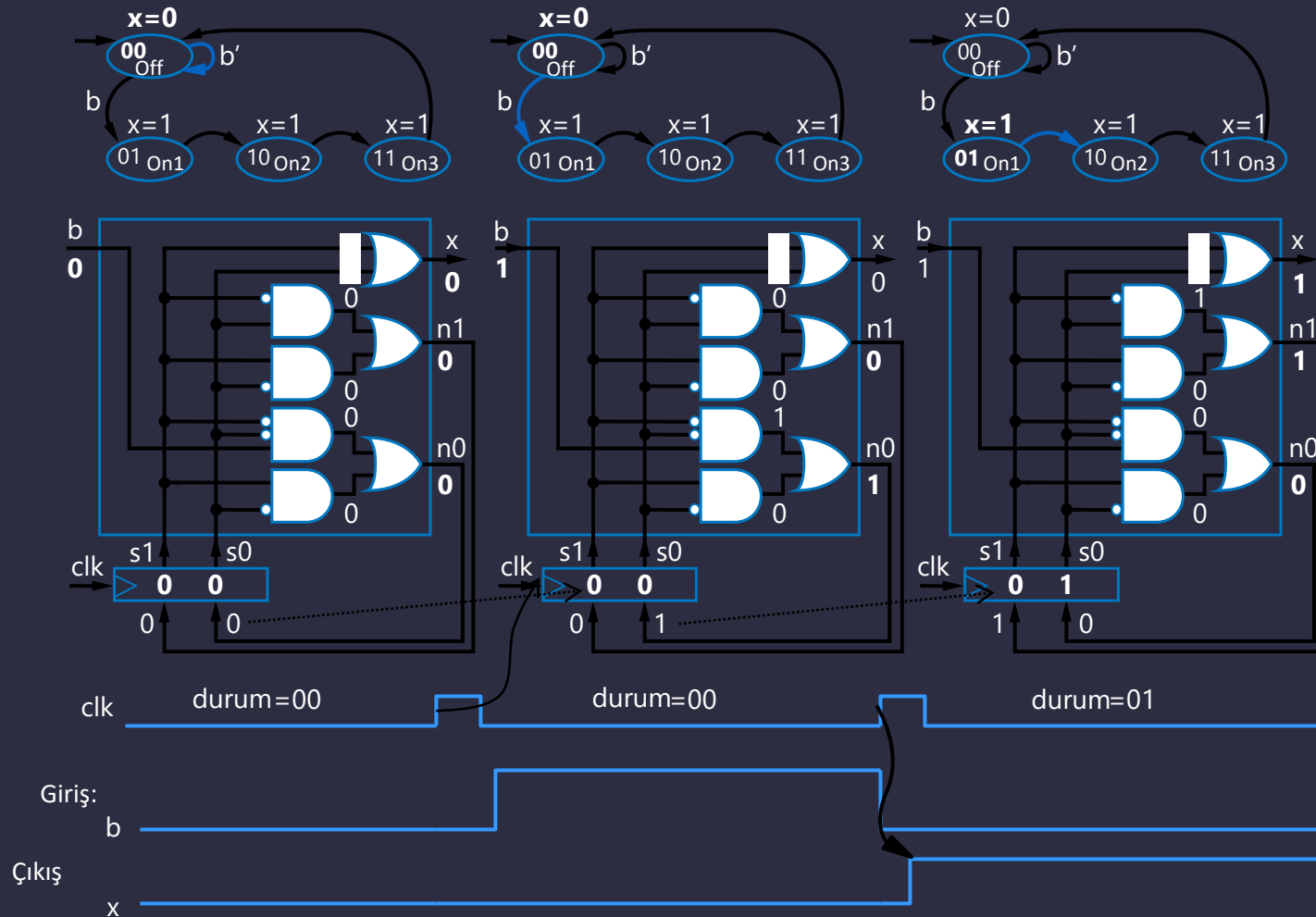
$$x = s1 \mid s0$$

$$n1 = s1's0 \mid s1s0'$$

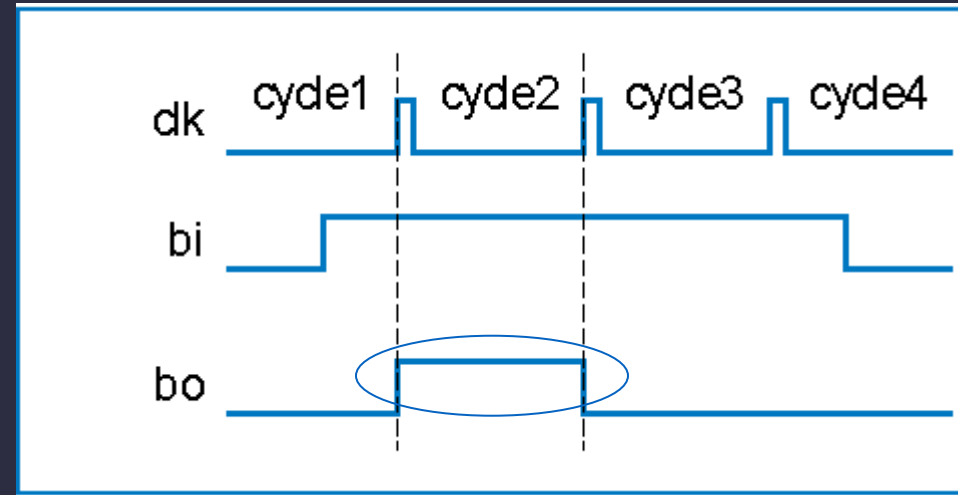
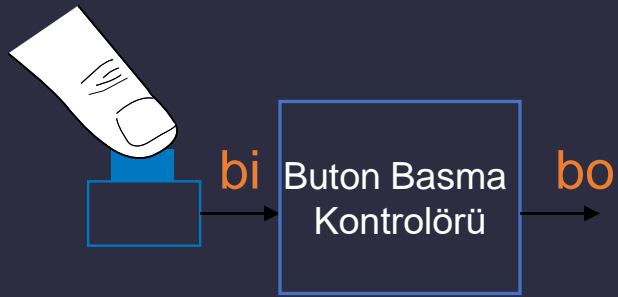
$$n0 = s1's0'b \mid s1s0'$$



FSM'nin İç Yapısı



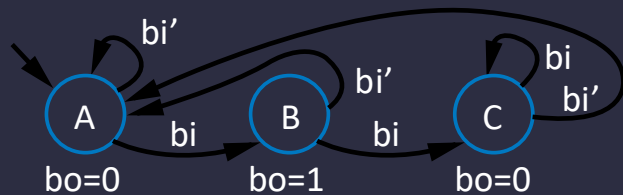
Buton Filtre Modülü Örneği



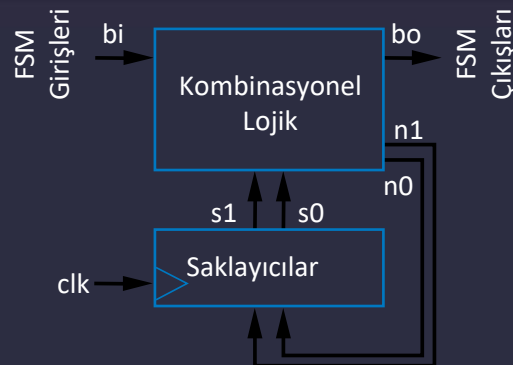
- Butona basıldığında sadece 1 cycle pulse üreten bir tasarım yapılmak isteniyor

Buton Filtre Modülü Örneği

FSM Girişler: bi; FSM çıkışlar: bo



Adım 1: FSM



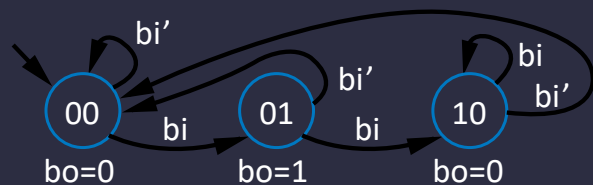
Adım 2: Mimari oluşturun

$$n1 = s1's0bi \mid s1s0bi$$

$$n0 = s1's0'bi$$

$$bo = s1's0bi' \mid s1's0bi = s1s0$$

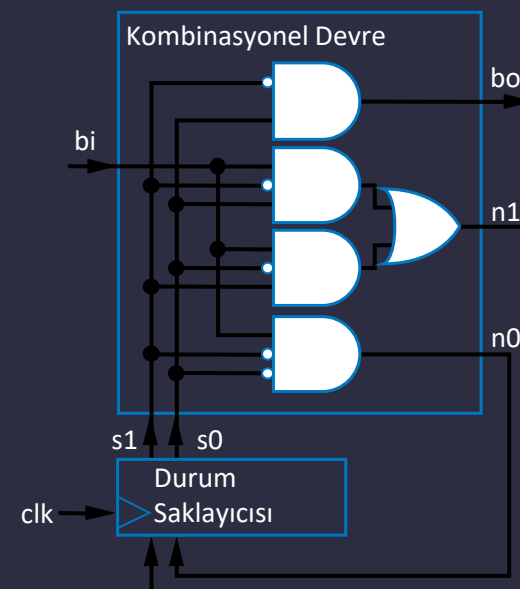
FSM giriş: bi; FSM çıkış: bo



Adım 3: Durum kodlaması

Girişler			Çıkışlar		
s1	s0	bi	n1	n0	bo
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	0	1
1	0	0	0	0	0
1	0	1	1	0	0
1	1	0	0	0	0
1	1	1	0	0	0

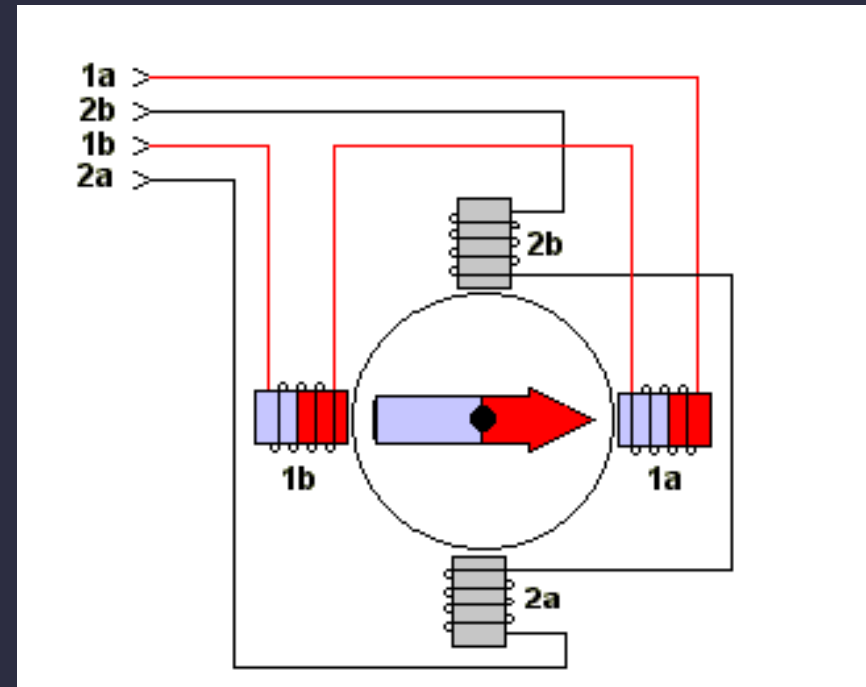
Adım 4: Durum Tablosu



Adım 5: Devre Oluşturun

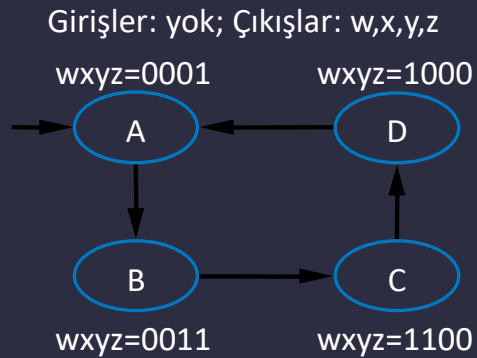
FSM Örneği

- 0001, 0011, 1100, 1000 ardışık olarak çıktı veren bir FSM tasarlayınız.
 - Örneğin step motor kontrolü için



FSM Örneği

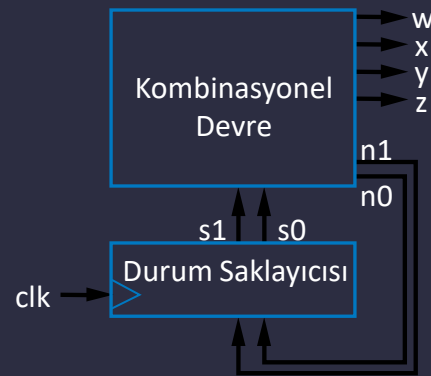
- 0001, 0011, 1100, 1000 ardışık olarak çıktı veren bir FSM tasarlayınız.
 - Örneğin step motor kontrolü için



Adım 1: FSM Oluşturma

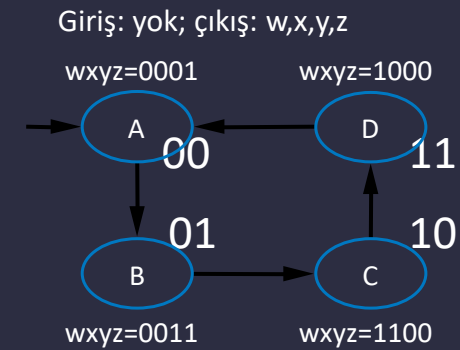
	Girişler		Çıktılar					
	s1	s0	w	x	y	z	n1	n0
A	0	0	0	0	0	1	0	1
B	0	1	0	0	1	1	1	0
C	1	0	1	1	0	0	1	1
D	1	1	1	0	0	0	0	0

Adım 4: Doğruluk Tablosu Oluşturma

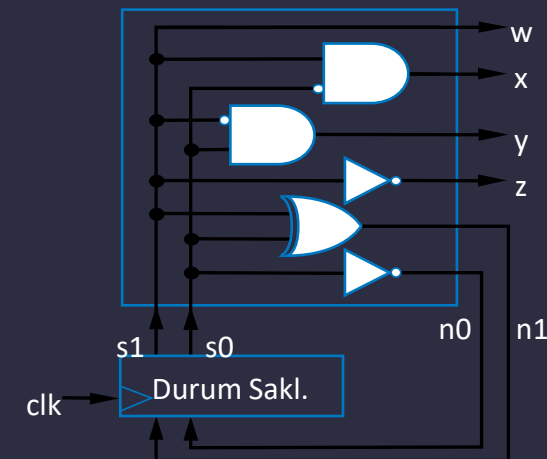


Adım 2: Mimari Oluşturma

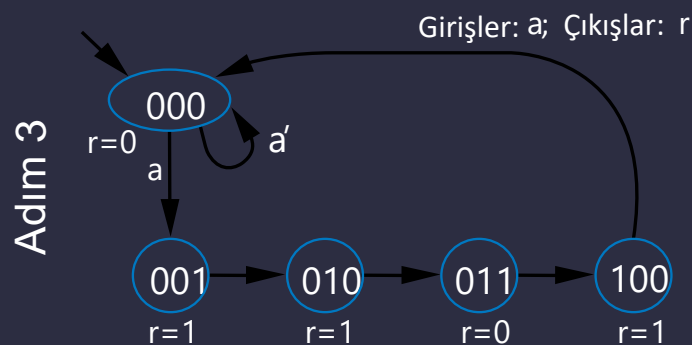
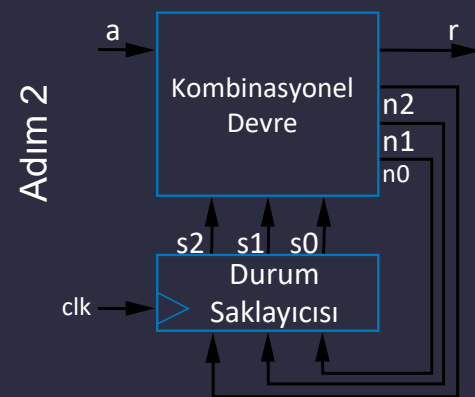
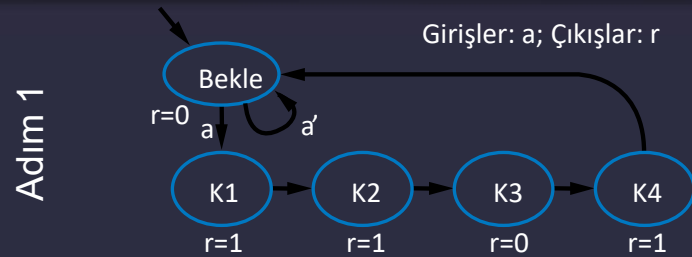
$$\begin{aligned}
 w &= s1 \\
 x &= s1s0' \\
 y &= s1's0 \\
 z &= s1' \\
 n1 &= s1 \text{ xor } s0 \\
 n0 &= s0'
 \end{aligned}$$



Adım 3: Durum Kodlaması



FSM Örneği, Araba Anahtarı

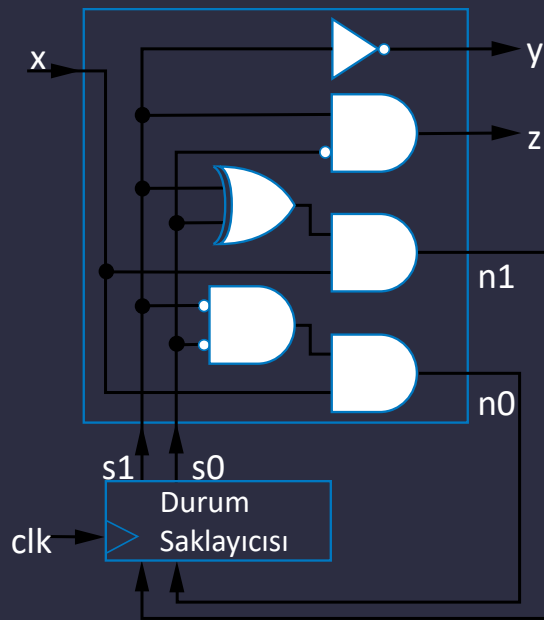


	Girişler				Çıktılar			
	s2	s1	s0	a	r	n2	n1	n0
Bekle	0	0	0	0	0	0	0	0
	0	0	0	1	0	0	0	1
K1	0	0	1	0	1	0	1	0
	0	0	1	1	1	0	1	0
K2	0	1	0	0	1	0	1	1
	0	1	0	1	1	0	1	1
K3	0	1	1	0	0	1	0	0
	0	1	1	1	0	1	0	0
K4	1	0	0	0	1	0	0	0
	1	0	0	1	1	0	0	0
-	1	0	1	0	0	0	0	0
	1	0	1	1	0	0	0	0
	1	1	0	0	0	0	0	0
	1	1	0	1	0	0	0	0
	1	1	1	0	0	0	0	0
	1	1	1	1	0	0	0	0

Adım 4

Devreden FSM'e Dönmek

Bu devre ne yapıyor?



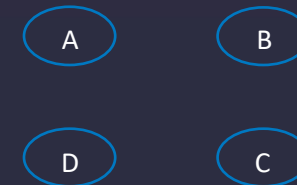
$$y = s1'$$

$$z = s1s0'$$

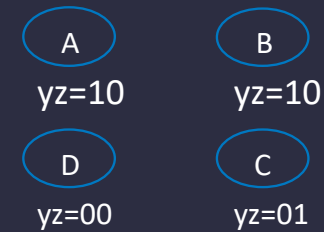
$$n1 = (s1 \text{ xor } s0)x$$

$$n0 = (s1' * s0')x$$

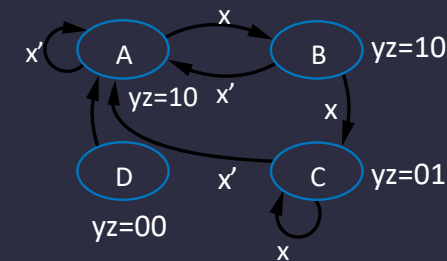
	Girişler			Çıkışlar			
	s1	s0	x	n1	n0	y	z
A	0	0	0	0	0	1	0
	0	0	1	0	1	1	0
B	0	1	0	0	0	1	0
	0	1	1	1	0	1	0
C	1	0	0	0	0	0	1
	1	0	1	1	0	0	1
D	1	1	0	0	0	0	0
	1	1	1	0	0	0	0



durumlar



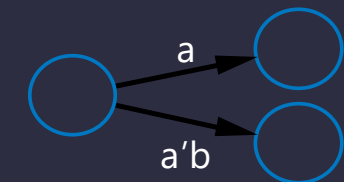
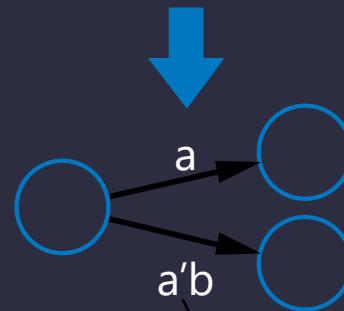
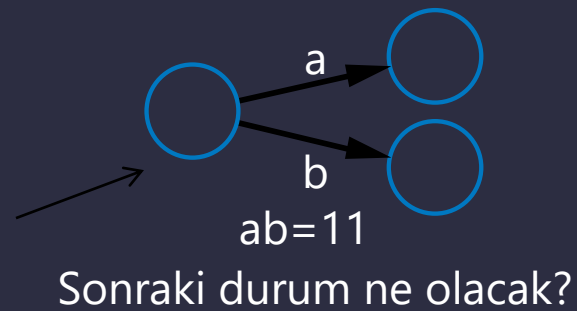
Durumlar ve çıkışlar



Durumlar, çıkışlar ve geçişler

FSM Tasarım Hataları

- *Durumlar arası geçişlerdeki koşullarda aynı anda tek bir geçiş koşulu doğru olmalıdır*



ab=00?
Sonraki durum ne olacak?

