



# BİLGİSAYAR ORGANİZASYONU ve TASARIMI

YRD. DOÇ. DR. FATİH KELEŞ

- ▶ Saklayıcılar (Registers)
- ▶ Register Transfer Dili (RTL)
- ▶ Register Transferi
- ▶ Ortak Yol (Bus) ve Bellek Transferi
- ▶ Aritmetik Mikroişlemleri
- ▶ Lojik Mikroişlemleri
- ▶ Öteleme Mikroişlemleri

- ▶ Saklayıcılar (Registers)
- ▶ Kaydırma Saklayıcıları (Shift Registers)
- ▶ Sayıcılar
- ▶ Bellek Birimleri

# Dijital Sistem

- ▶ Belirli bir bilgi işlem görevini yerine getiren donanımsal modüllerin bağlanması ile oluşturulur.
- ▶ Donanımsal Modüller
  - Saklayıcılar (registers)
  - Kod Çözüçüler (decoders)
  - Aritmetik Devreler
  - Kontrol Lojik Devreleri
- ▶ Modüller içerdikleri saklayıcılar ve icra ettikleri işlemler (mikroişlemler) ile tanımlanırlar.

# Dijital Bilgisayar

Donanımsal Organizasyonu karakterize eden unsurlar:

- ▶ İçerdikleri saklayıcılar (registers)
- ▶ İcra ettikleri işlemler (mikroişlemler)
  - Tipik olarak saklayıcılar arasında transfer edilen ikili bilgi ve saklayıcılar üzerinde gerçekleşen işlemler
- ▶ Mikroişlem dizinini başlatan ve yönlendiren kontrol işaretleri üretimi

# Saklayıcılar (Registers)

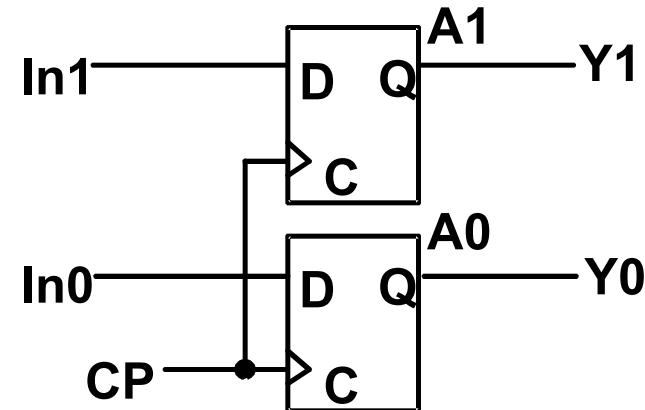
- ▶ İkili data saklayan elemanlardan oluşan bir data tutma bloğu olarak düşünülebilir.
- ▶ Saklayıcı, bir Ardışıl Lojik Devre yapısıdır.
- ▶ Data tutma işlemi yanı sıra, data transferi ve data işlemlerini de yerine getirir.
- ▶ Bir saklayıcı, bir bitlik bilgiyi saklama kapasitesine sahip flip-foplardan oluşur.
- ▶ n-bit saklayıcı, n adet flip-flop içerir.

# Saklayıcılar (Registers)

- ▶ Flip-foplara ek olarak datayı işleme amaçları oluşturmak için kullanılan kapı dizilerine sahip olabilir.
- ▶ Daha geniş olarak saklayıcılar kapılar ve flip-foplardan oluşur.
- ▶ Flip-foplardan ikili bilgiyi tutar, kapılar yeni bilginin saklayıcı içerisinde aktarılıp aktarılmadığını kontrol eder.

# Örnek: 2-bit Saklayıcı

- ▶ Durum sayısı?
- ▶ Giriş kombinezonu sayısı?
- ▶ Çıkış fonksiyonu?
- ▶ Bir sonraki durum fonksiyonu?



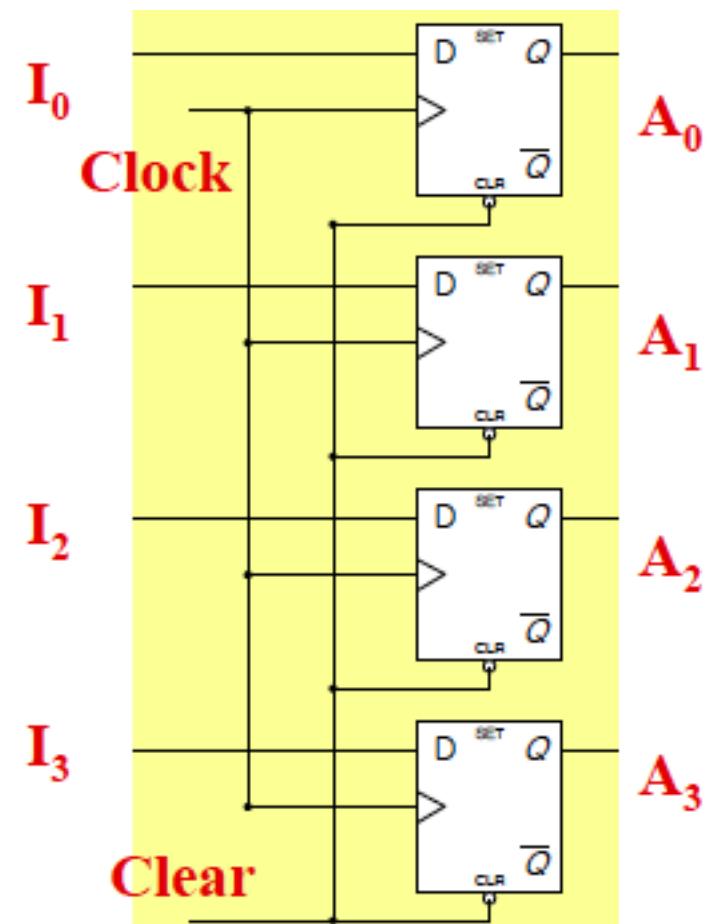
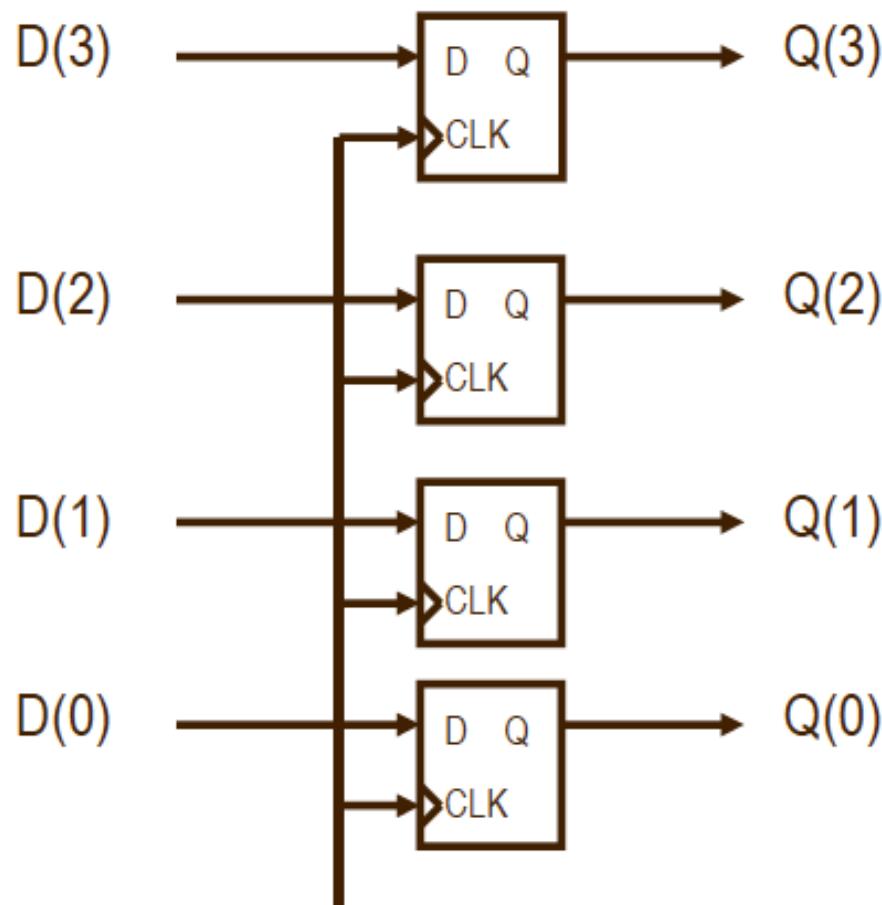
State Table:

| Current State<br>A1 A0 | Next State<br>A1(t+1) A0(t+1)<br>For In1 In0 =<br>00 01 10 11 | Output<br>(=A1 A0)<br>Y1 Y0 |
|------------------------|---|-----------------------------|
| 0 0                    | 00 01 10 11   | 0 0                         |
| 0 1                    | 00 01 10 11   | 0 1                         |
| 1 0                    | 00 01 10 11   | 1 0                         |
| 1 1                    | 00 01 10 11   | 1 1                         |

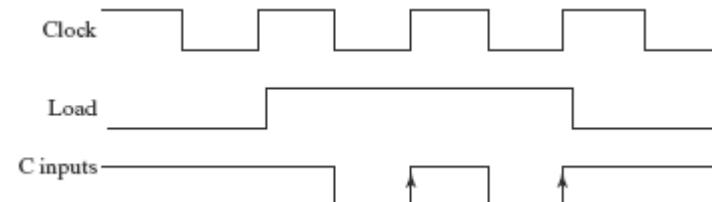
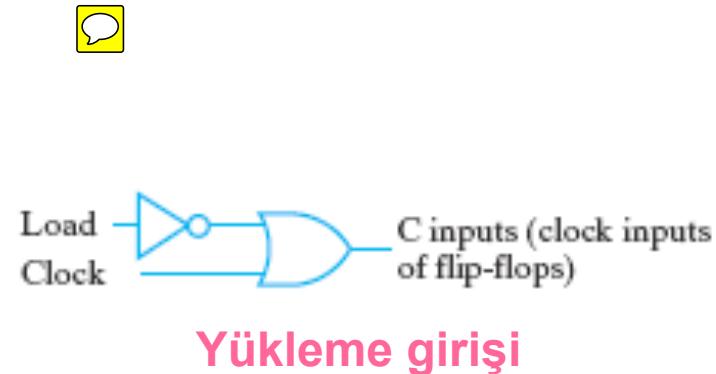
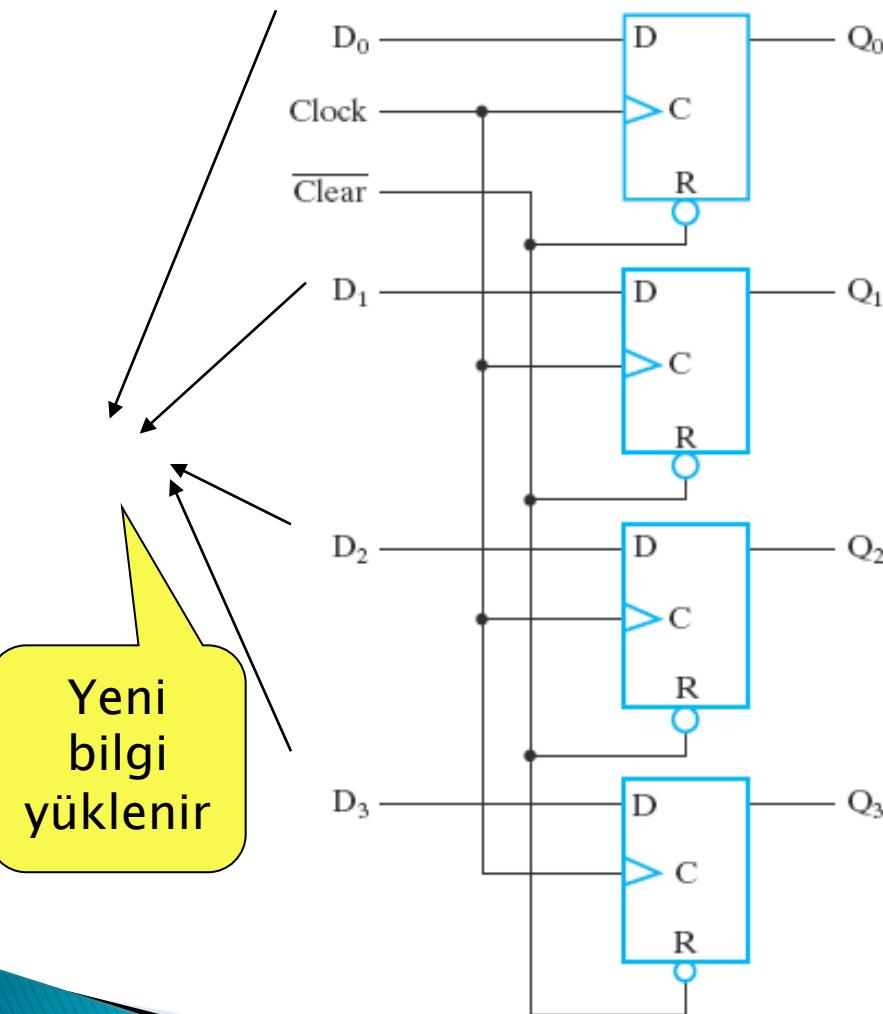
# **Saklayıcı Tasarım Modelleri**

- ▶ N artıkça durum ve giriş kombinezonu sayısı arttığından durum diyagramı/durum tablosu modeli ile tasarım zorlaşabilir.
- ▶ Tasarım için diğer metodlar?
  - Saklayıcı f.f.larına önceden belirlenmiş KLD devreleri bağlamak
    - Örnek: İleri sayma için, saklayıcı ff.larına artıran KLD yapısını bağlanır.
    - Hücresel Tasarım Modeli : Durum tablosu/durum diyagramı modelini kullanarak hücresel tasarım yapıp hücreleri bir register halinde birleştirmek
    - 1-bit hücrenin sadece 2 durumu vardır:
      - Çıkış genellikle bir durum değişkenidir.

# 4-bit Saklayıcı



# 4-bit Saklayıcı - Yükleme Kontrolü



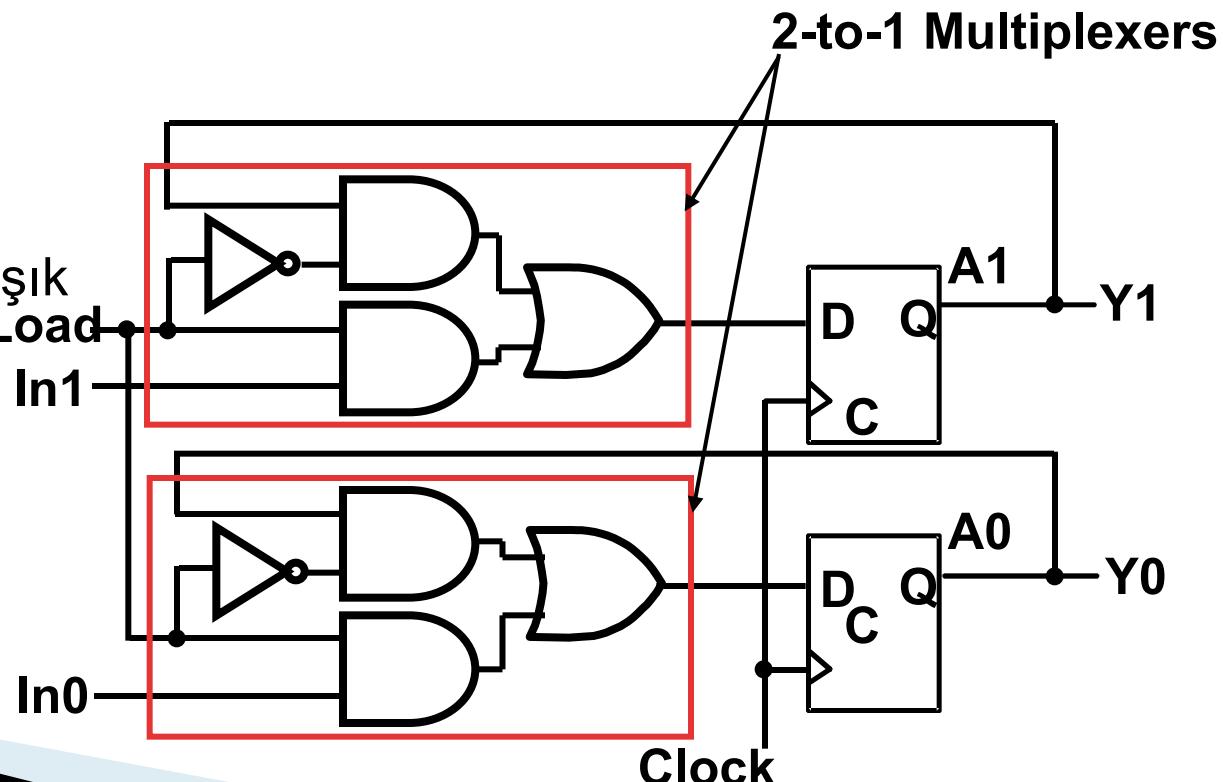
Zamanlama Şeması

# Yükleme Kontrollü Saklayıcı

- ▶ Yüklemenin daha güvenli yapılabilmesinin yolu  
Saklayıcı içeriğini değiştirmek için bir yükleme kontrol devresi  
kullanmaktadır.

Örnek: 2-bit saklayıcı

- ▶ Load =0 için,  
İçerik korunur
- ▶ Load = 1 için,  
yeni içerik yüklenir
- ▶ Donanım daha karmaşık  
ancak zamanlama  
problemi yok



# Paralel Yüklemeli Saklayıcılar

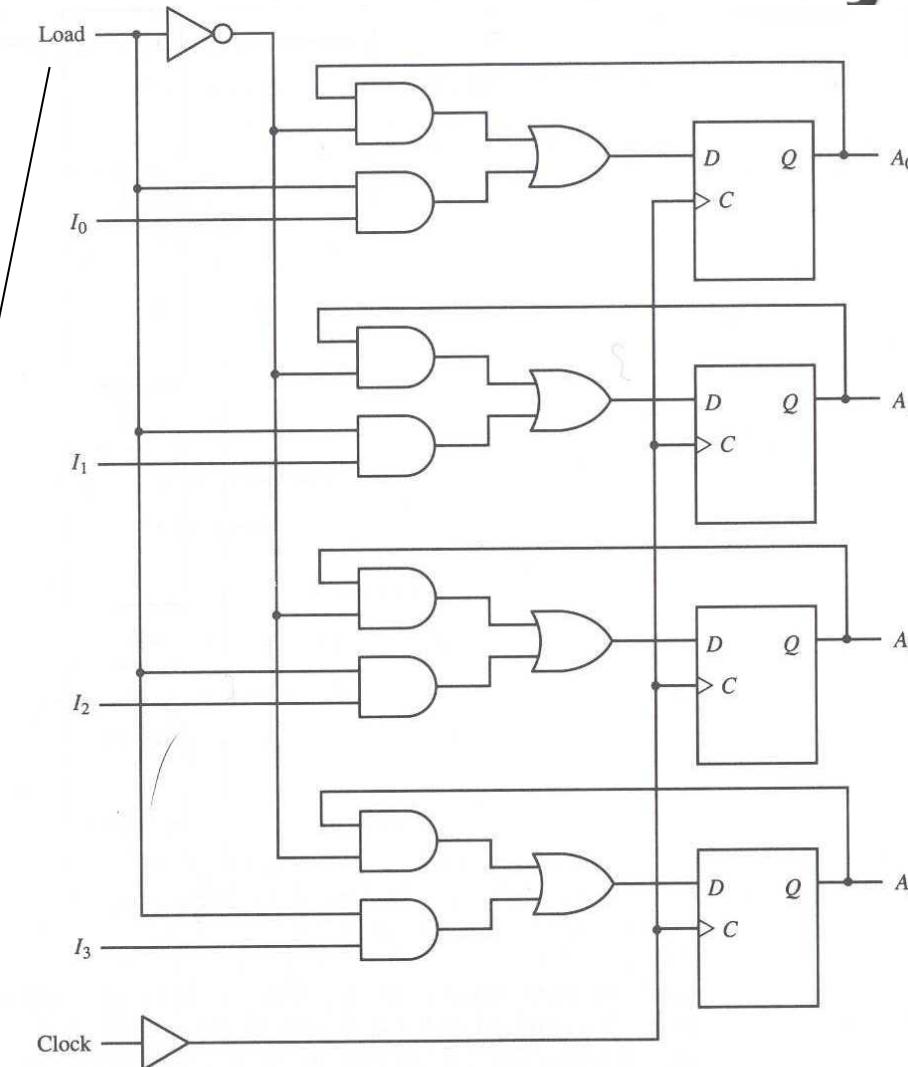
- ▶ Girişlerden saklayıcılara bilgi aktarımı, tek bir saat işaretti girişinde tüm bitler için aynı anda yapılır.
- ▶ Yükleme girişi 1 olduğunda, 4-bitlik bir saklayıcıda 4 girişteki veri bir sonraki saat vuruşunun yukarı çıkışında saklayıcıya aktarılır.
- ▶ Yükleme girişi 0 olduğunda, veri girişleri ayrıılır ve D flip-floplarında mevcut durum korunur.



# Paralel Yüklemeli Saklayıcılar

Load(Yükle)=0  
olduğunda  
mevcut durum  
korunur.

Load(Yükle)=1  
olduğunda  
paralel olarak  
yeni bilgi  
yüklenir.

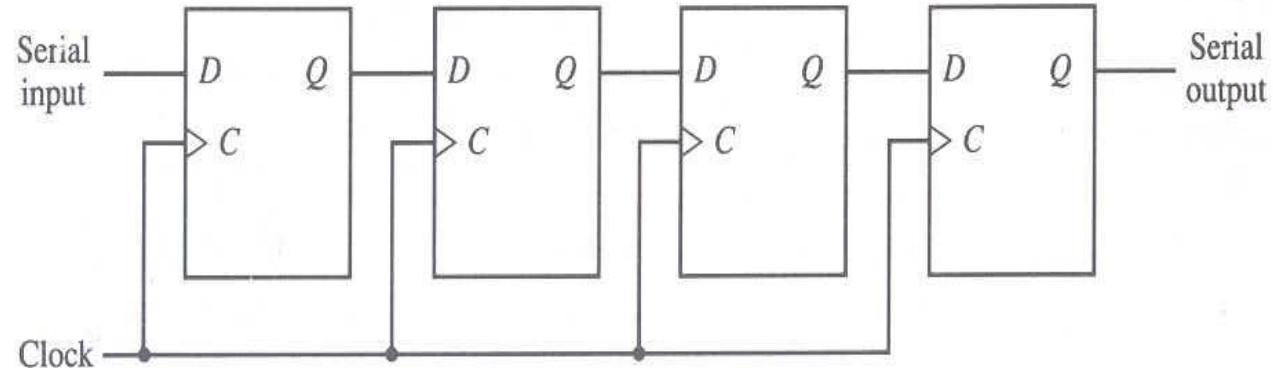


# Kaydırma Saklayıcıları

- ▶ İkili bilgiyi bir veya daha fazla yöne kaydırılan saklayıcılara **kaydırma saklayıcıları** denir.
- ▶ Bir kaydırma saklayıcısı, bir flip-flopun çıkışı bir sonraki flip-flopun girişine bağlanarak oluşturulur.
- ▶ Saat vuruşları ortaktır.

# Kaydırma Saklayıcıları

0110 verisini  
yüklemek  
için



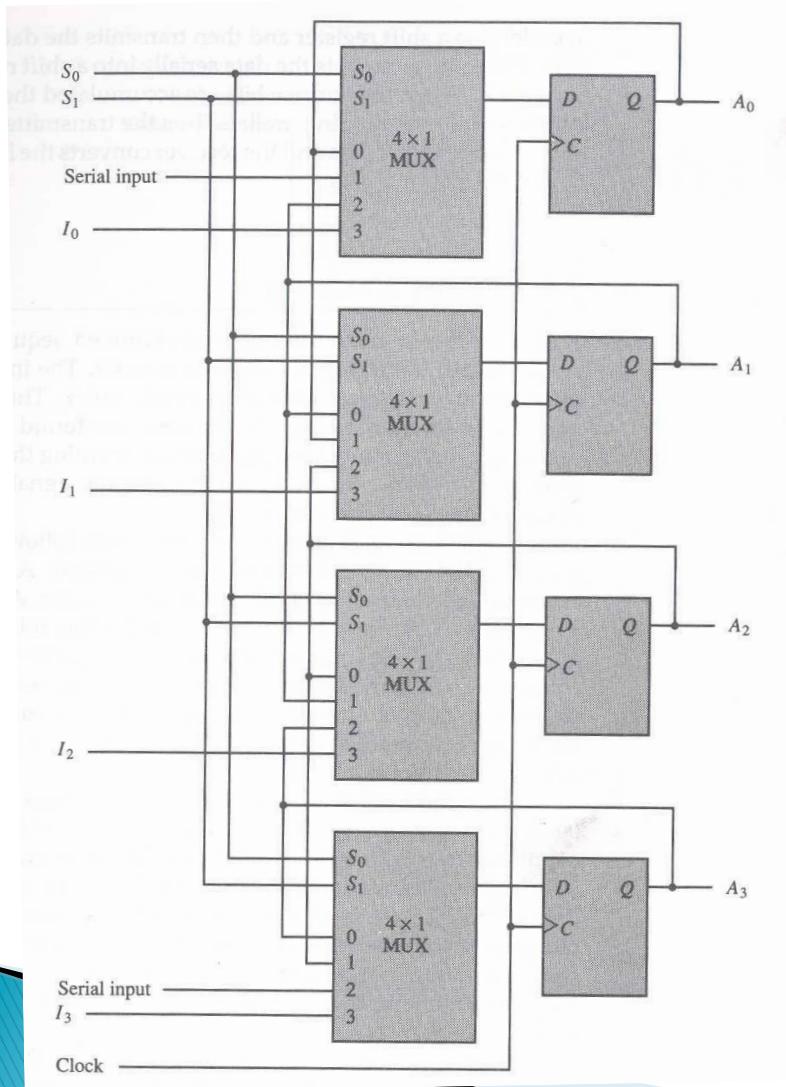
|     |          |          |          |          |
|-----|----------|----------|----------|----------|
| T0: | veri yok | veri yok | veri yok | veri yok |
| T1: | <b>0</b> | veri yok | veri yok | veri yok |
| T2: | <b>1</b> | <b>0</b> | veri yok | veri yok |
| T3: | <b>1</b> | <b>1</b> | <b>0</b> | veri yok |
| T4: | <b>0</b> | <b>1</b> | <b>1</b> | <b>0</b> |

# Paralel Yüklemeli Çift Yönlü Kaydırma Saklayıcıları

- ▶ Yalnızca tek bir yönde kaydırma özelliğine sahip saklayıcılara **tek yönlü**, her iki yönde de kaydırma özelliğine sahip saklayıcılara **çift yönlü** kaydırma saklayıcıları denir.
- ▶ Fonksiyon Çizelgesi

| $S_1$ | $S_0$ | Saklayıcı İşlemi |
|-------|-------|------------------|
| 0     | 0     | Değişiklik Yok   |
| 0     | 1     | Sağa Kaydırma    |
| 1     | 0     | Sola Kaydırma    |
| 1     | 1     | Paralel Yükleme  |

# Paralel Yüklemeli Çift Yönlü Kaydırma Saklayıcıları



| S1 | S0 |
|----|----|
| 0  | 0  |
| 0  | 1  |
| 1  | 0  |
| 1  | 1  |

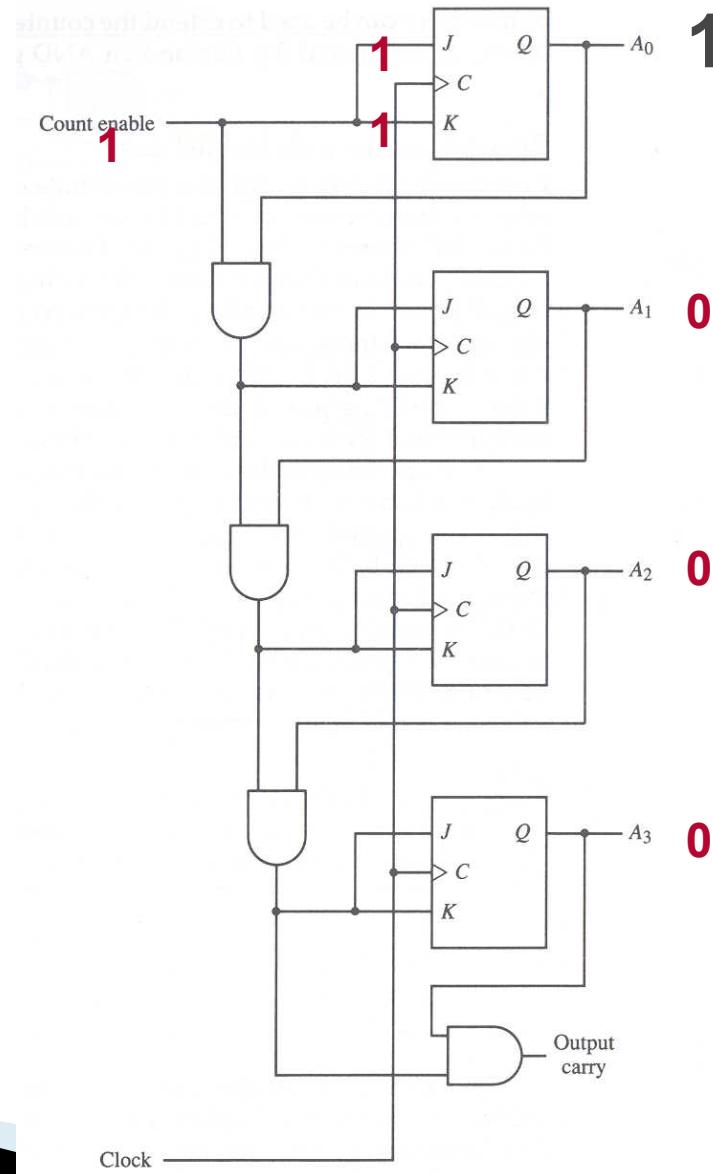
**Saklayıcı İşlemi**  
Değişiklik Yok  
Sağa Kaydırma  
Sola Kaydırma  
Paralel Yükleme

$A_0 \ A_1 \ A_2 \ A_3$   
 $0 \ A_0 \ A_1 \ A_2 \text{ sağa (aşağı)}$   
 $A_1 \ A_2 \ A_3 \ 0 \text{ sola (yukarı)}$

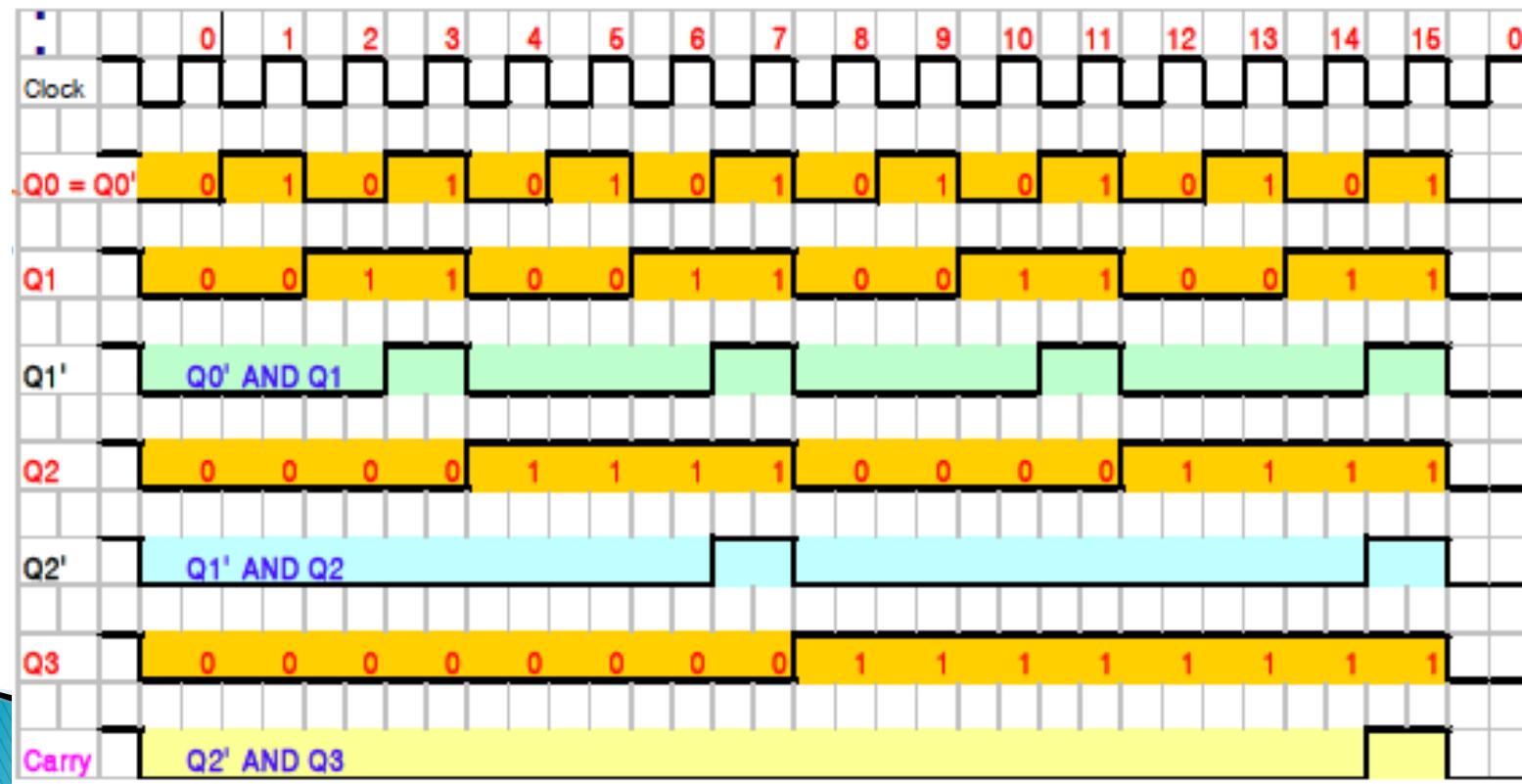
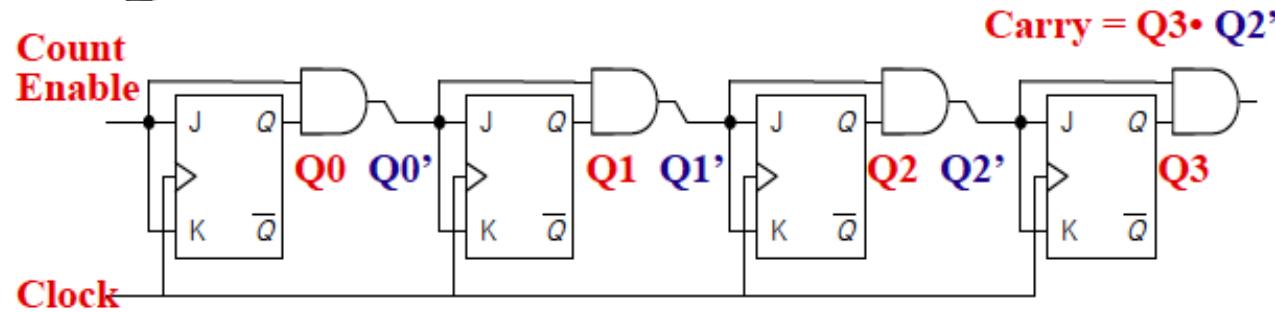
# İkili Sayıcılar

- ▶ JK flip-flopunda, J ve K girişlerinin her ikisi de 1 ise ve saat işaretini pozitif geçişte tümleyen alınır.
- ▶ İkisi de 0 ise, durum değişmez.

# İkili Sayıcılar

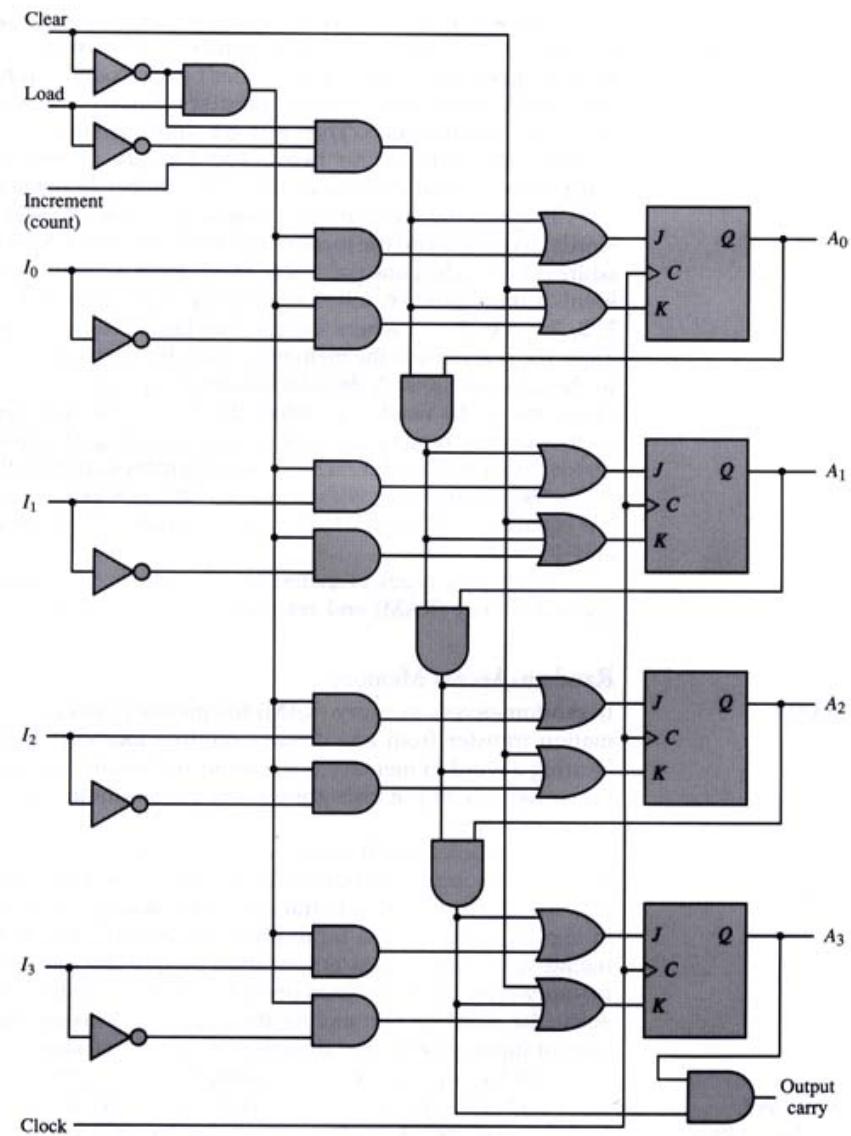


# İkili Sayıcılar



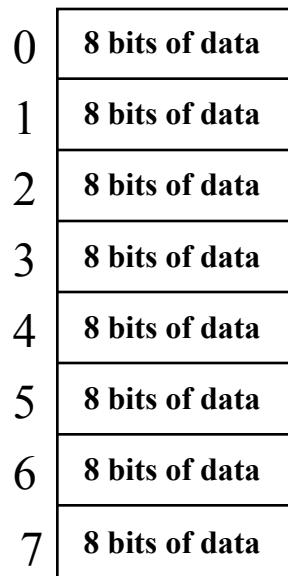
# Paralel Yüklemeli Saatli Silmeli 4 Bit İkili Sayıcı

| CL | LD | INC | İşlem                   |
|----|----|-----|-------------------------|
| 0  | 0  | 0   | Değişiklik yok          |
| 0  | 0  | 1   | Sayma 1 artar           |
| 0  | 1  | x   | Yükleme girişleri aktif |
| 1  | x  | x   | Cıkışlar 0'la silinir   |



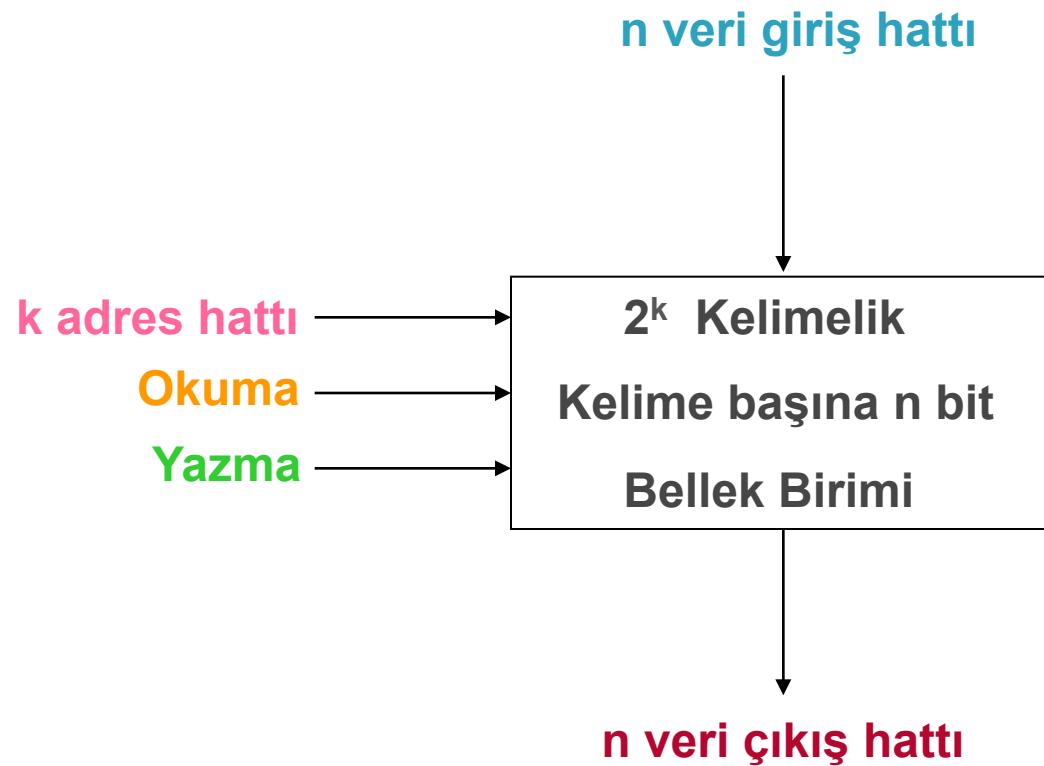
# Bellek Birimi

- ▶ Bilgiyi saklayabilen saklayıcılarından oluşan birimdir. Bellek **kelime** olarak tanımlanan ikili bilgileri bit grupları halinde saklar.
- ▶ 8 bitten oluşan gruplara **byte** denir.



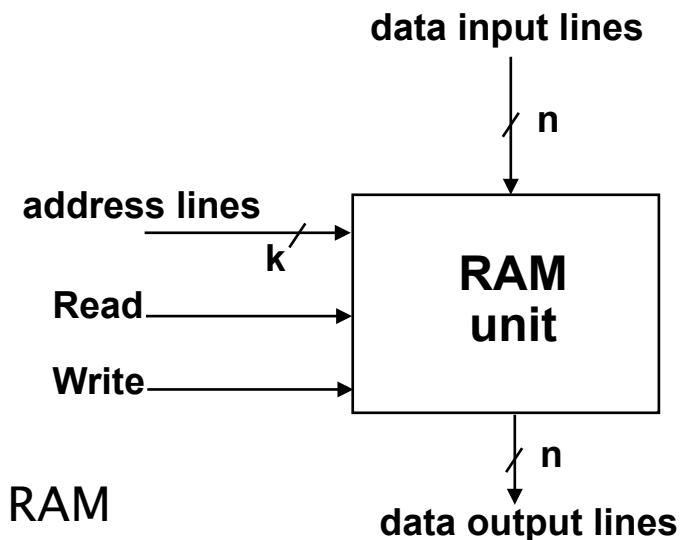
|     |          |
|-----|----------|
| K   | $2^{10}$ |
| M   | $2^{20}$ |
| G   | $2^{30}$ |
| 4K  | $2^{12}$ |
| 64K | $2^{16}$ |
| 16M | $2^{24}$ |

# Bellek Birimi



# BELLEK (RAM)

- ▶ Bellek (RAM) bir dizi saklayıcı içeren ardışıl lojik devre olarak düşünülebilir.
- ▶ Bu saklayıcılar bellek kelimesini tutarlar
- ▶  $r$  saklayıcının herbiri bir adresle gösterilir.
- ▶ Her saklayıcı  $n$  bitlik bir data tutar (kelime)
- ▶ Kabul: RAM  $n$ -bitlik  $r$  saklayıcı =  $2^k$  adet kelime içeriyor olsun. RAM aşağıda verilen bileşenlere sahip olmalıdır:
  - $n$  data giriş hattı
  - $n$  data çıkış hattı
  - $k$  adres hattı
  - Read kontrol işareteti
  - Write kontrol işareteti



- $2^k \times n$ -bit RAM
- Ör: adr 5bit, data 8bit
- 32 x 8-bit RAM

# Bellek Birimi

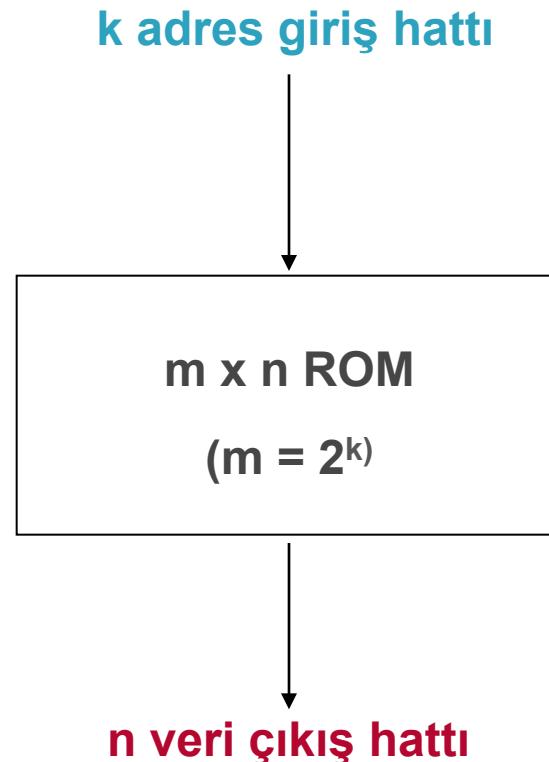
## ▶ Yazma işlemi

- İstenen kelimenin ikili adresini adres yoluna yaz.
- Belleğe yazılması istenilen veri bitlerini veri yoluna yaz.
- Yaz girişini çalıştır.

## ▶ Okuma işlemi

- İstenen kelimenin ikili adresini adres yoluna yaz.
- Oku girişini çalıştır.

# Yalnız Okunabilir Bellek(ROM)



ROM: decoder + OR  
Kombinasyonel tümleşik devre

# ROM Çeşitleri

- ▶ PROM, programlanabilir ROM
- ▶ EEPROM veya EPROM, silinebilen ROM.  
EPROM sigortaları özel ultraviyole ışık altında belli bir süre bırakılırsa, kısa dalga radyasyonu sigorta olarak kullanılan iş kapılarını devre dışı bırakır.