

# Bilgisayar Bilimine Giriş

13. Baskı, Global Edition

## Bölüm 1

Veri depolama

# Bölüm 1: Veri depolama

- 1.1 Bitler ve Depolanmaları
- 1.2 Ana Bellek
- 1.3 Yığın Depolama
- 1.4 Bilginin Bit Desenleri Cinsinden Gösterimi
- 1.5 İkili Sistem

# Bölüm 1: Veri depolama (devamı)

- 1.6 Tam Sayıların Depolanması
- 1.7 Kesirli Sayıların Depolanması
- 1.8 Veri ve Programlama
- 1.9 Veri Sıkıştırma
- 1.10 İletişim Hataları

# 1.1 Bitler ve Depolanmaları

- **Bit:** İkilik Rakamlardır(0 veya 1)
- Bit desenleri bilgiyi ifade etmek için kullanılırlar
  - Sayılar
  - Harfler
  - Resimler
  - Ses
  - Ve diğer veriler

# Boolean İşlemleri

- **Boolean İşlemi:** Bir veya daha fazla doğru/yanlış değerini yöneten işlemdir
- Özel işlemler
  - AND (Ve)
  - OR (Veya)
  - XOR (Ayricalıklı OR)
  - NOT (Değil)

# Şekil 1.1 Boolean işlemleri AND, OR ve XOR(Özel OR) ‘un olası giriş ve çıkış değerleri

## AND(Ve) işlemi

$$\begin{array}{r} 0 \\ \text{AND } 0 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 0 \\ \text{AND } 1 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 1 \\ \text{AND } 0 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 1 \\ \text{AND } 1 \\ \hline 1 \end{array}$$

## OR(Veya) işlemi

$$\begin{array}{r} 0 \\ \text{OR } 0 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 0 \\ \text{OR } 1 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 1 \\ \text{OR } 0 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 1 \\ \text{OR } 1 \\ \hline 1 \end{array}$$

## XOR(Özel OR)

$$\begin{array}{r} 0 \\ \text{XOR } 0 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 0 \\ \text{XOR } 1 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 1 \\ \text{XOR } 0 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 1 \\ \text{XOR } 1 \\ \hline 0 \end{array}$$

# Kapılar

- **Kapı:** Boolean işlemini hesaplayan bir aygıttır
  - Genelde transistör denilen küçük elektronik devreler şeklinde kullanılırlar
  - Çeşitli teknolojilerle etkileşimde bulunabilir, inşa edilebilirler
  - Bilgisayarların oluşturulduğu yapı bloklarını sağlamaktadırlar

# Şekil 1.2 AND,OR,XOR ve NOT kapılarının giriş ve çıkış değerleriyle beraber gösterimi



Giriş Değerleri	Çıkış Değeri
0 0	0
0 1	0
1 0	0
1 1	1



Giriş Değerleri	Çıkış Değeri
0 0	0
0 1	1
1 0	1
1 1	1



Giriş Değerleri	Çıkış Değeri
0 0	0
0 1	1
1 0	1
1 1	0



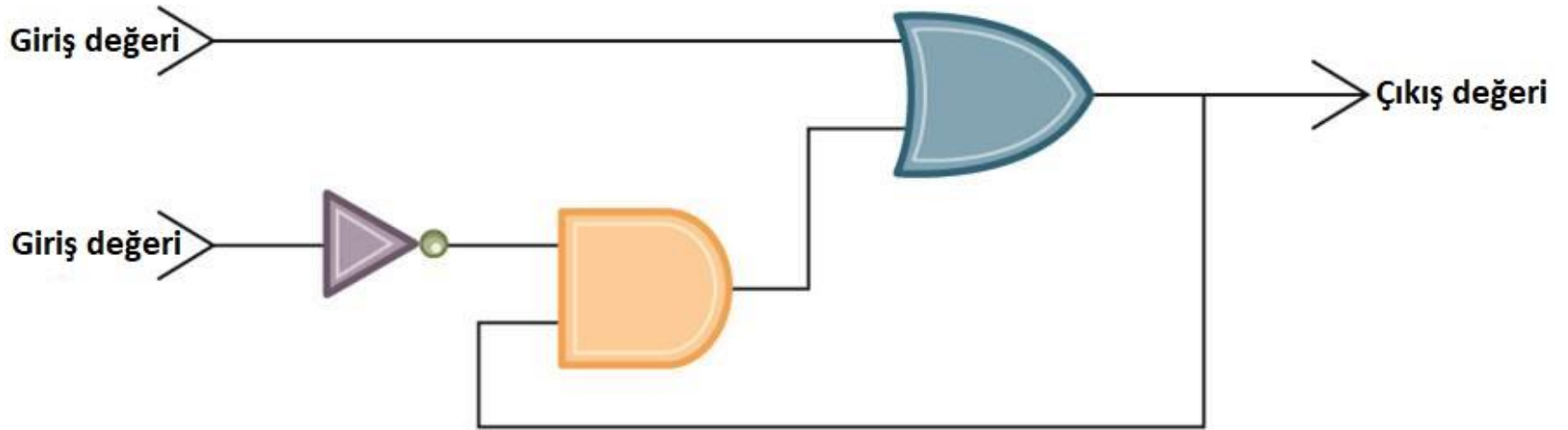
Giriş Değeri	Çıkış Değeri
0	1
1	0



# Flip-floplar

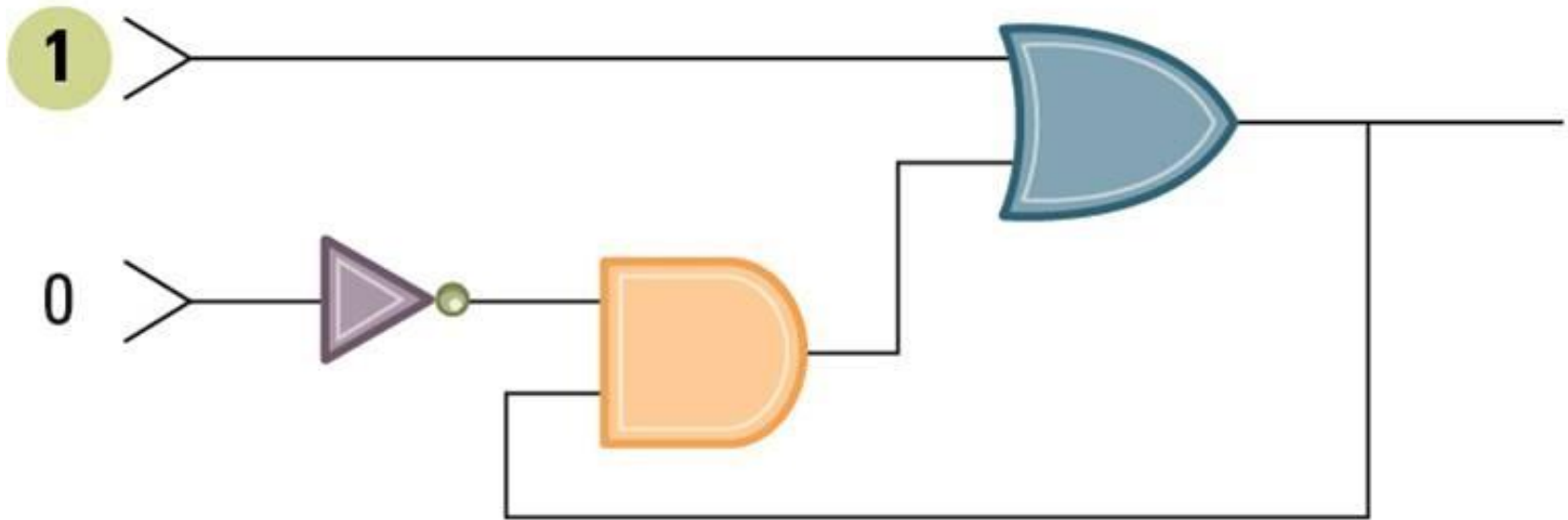
- Bilgisayar hafızasının temel birimi olan kapılardan oluşturulmuş devrelerdir
  - Depolanan değeri 1'e dönüştürmek için bir giriş değeri kullanılır
  - Depolanan değeri 0'a dönüştürmek için bir giriş değeri kullanılır
  - İki giriş değeri de 0 ise son depolanan değer korunur

## Şekil 1.3 Basit bir flip-flop devresi



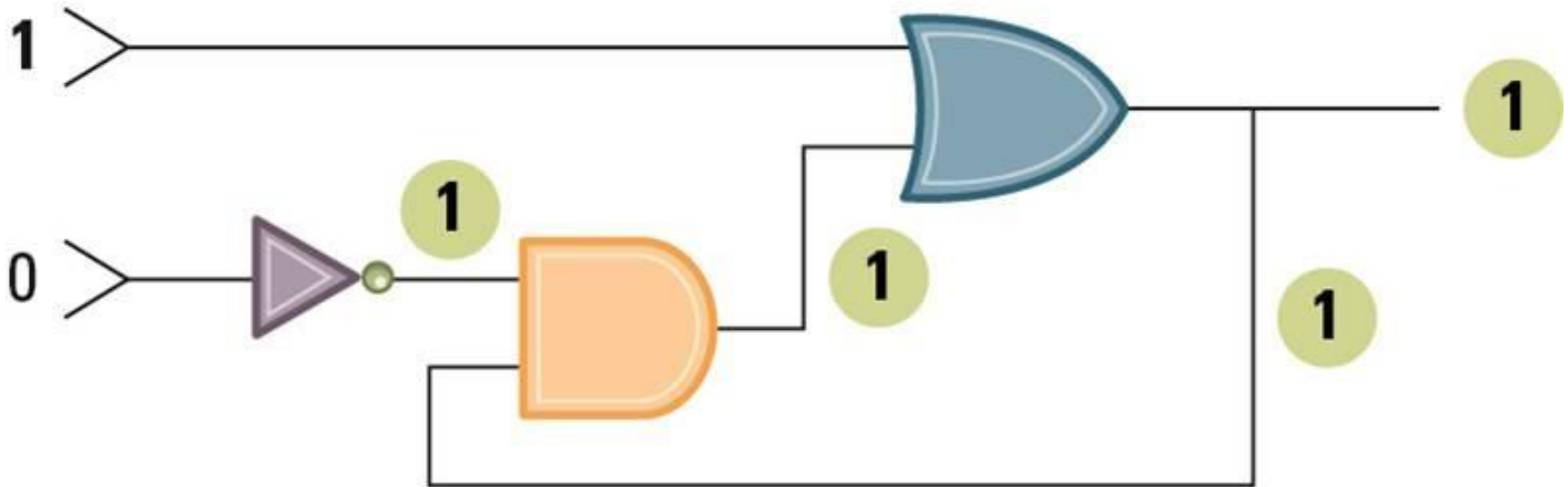
# Şekil 1.4 Bir flip-flop'un çıkış değerini 1'e ayarlama

a. İlk olarak üstteki giriş değerine 1 atanır.



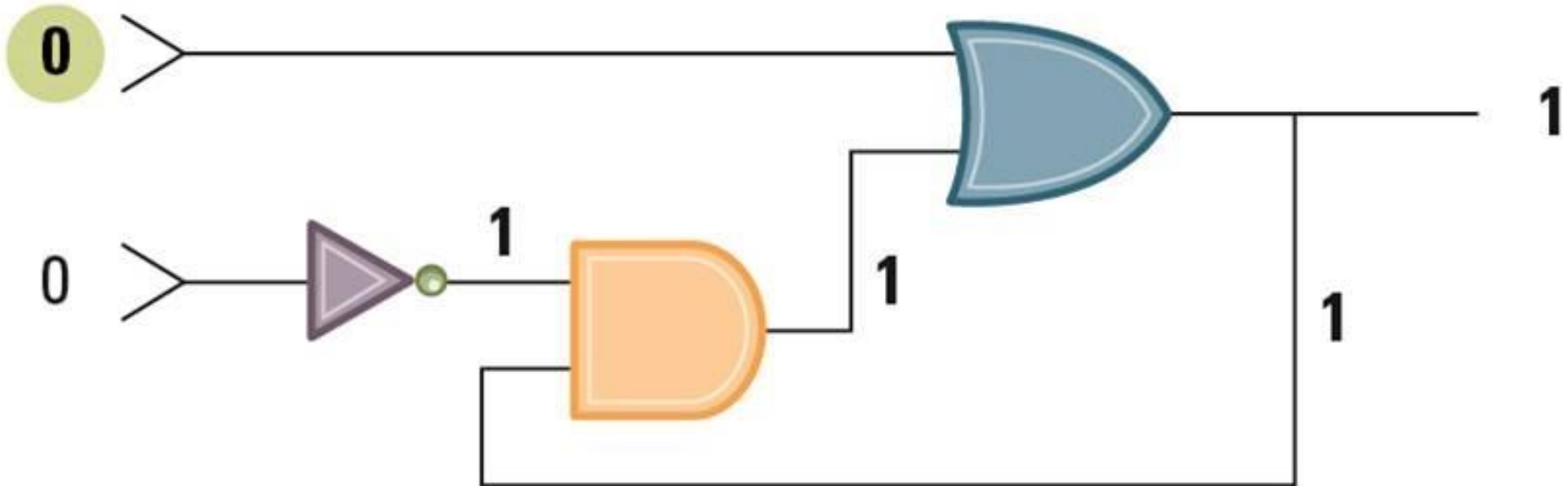
## Şekil 1.4 Bir flip-flop'un çıkış değerini 1'e ayarlama (devamı)

**b.** Bu, OR kapısının çıkışının 1 olmasına ve buna bağlı olarak AND kapısının da çıkışının 1 olmasına sebep olur

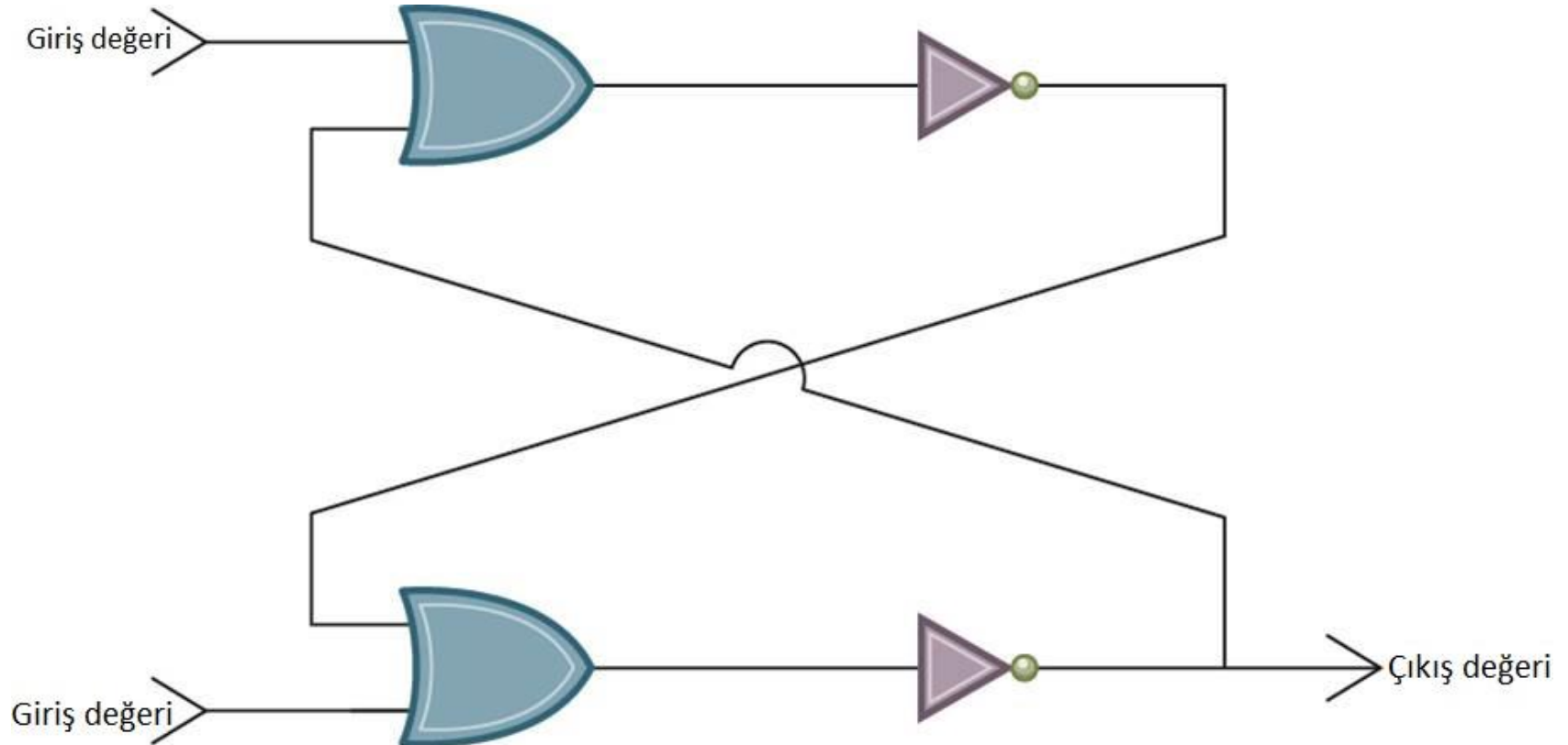


## Şekil 1.4 Bir flip-flop'un çıkış değerini 1'e ayarlama (devamı)

c.Son olarak, AND kapısından gelen 1 değeri, OR kapısındaki değeri 0'a döndüğünde bile sonucun değişmesini önler.



# Şekil 1.5 Bir flip-flop oluşturma'nın bir diğer yolu



# Onaltılık(Hexadecimal) Sayı Gösterimi

- **Onaltılık Sayı Gösterimi:** Uzun bit desenlerini göstermenin kısa bir yoludur.
  - Bir deseni, her biri 4 bitten oluşan gruplara böler
  - Her grup bir sembolle ifade edilir
- Örnek: 10110101 => 0xB5 şeklinde

# Şekil 1.6 Onaltılık(Hexadecimal) Sayı Kodlama Biçimi

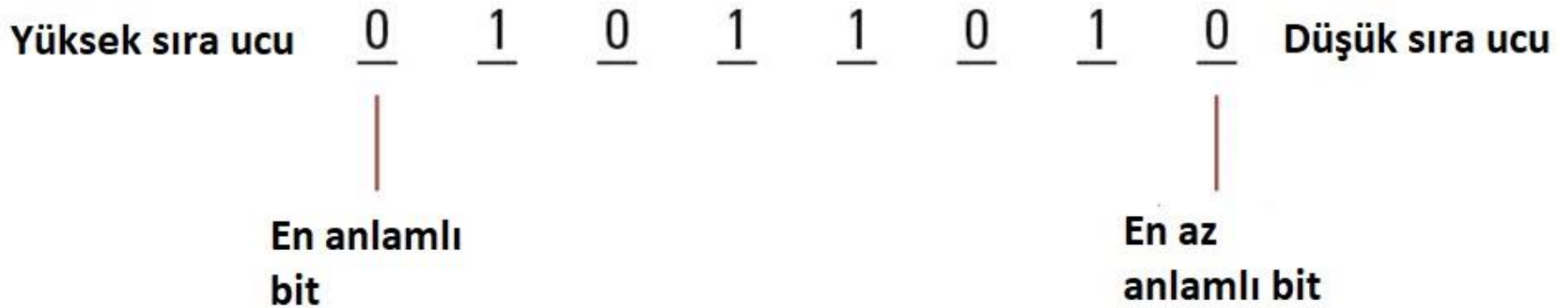
Bit gösterimi	Onaltılık gösterim
0000	0x0
0001	0x1
0010	0x2
0011	0x3
0100	0x4
0101	0x5
0110	0x6
0111	0x7
1000	0x8
1001	0x9
1010	0xA
1011	0xB
1100	0xC
1101	0xD
1110	0xE
1111	0xF



## 1.2 Ana Bellek

- **Hücre:** Ana belleğin bir birimidir(Genellikle bir **byte**'a eşit olan 8 bittir)
  - **En anlamlı bit:** Sol uçtaki bittir (yüksek sıra ucu)
  - **En az anlamlı bit:** Sağ uçtaki bittir (düşük sıra ucu)

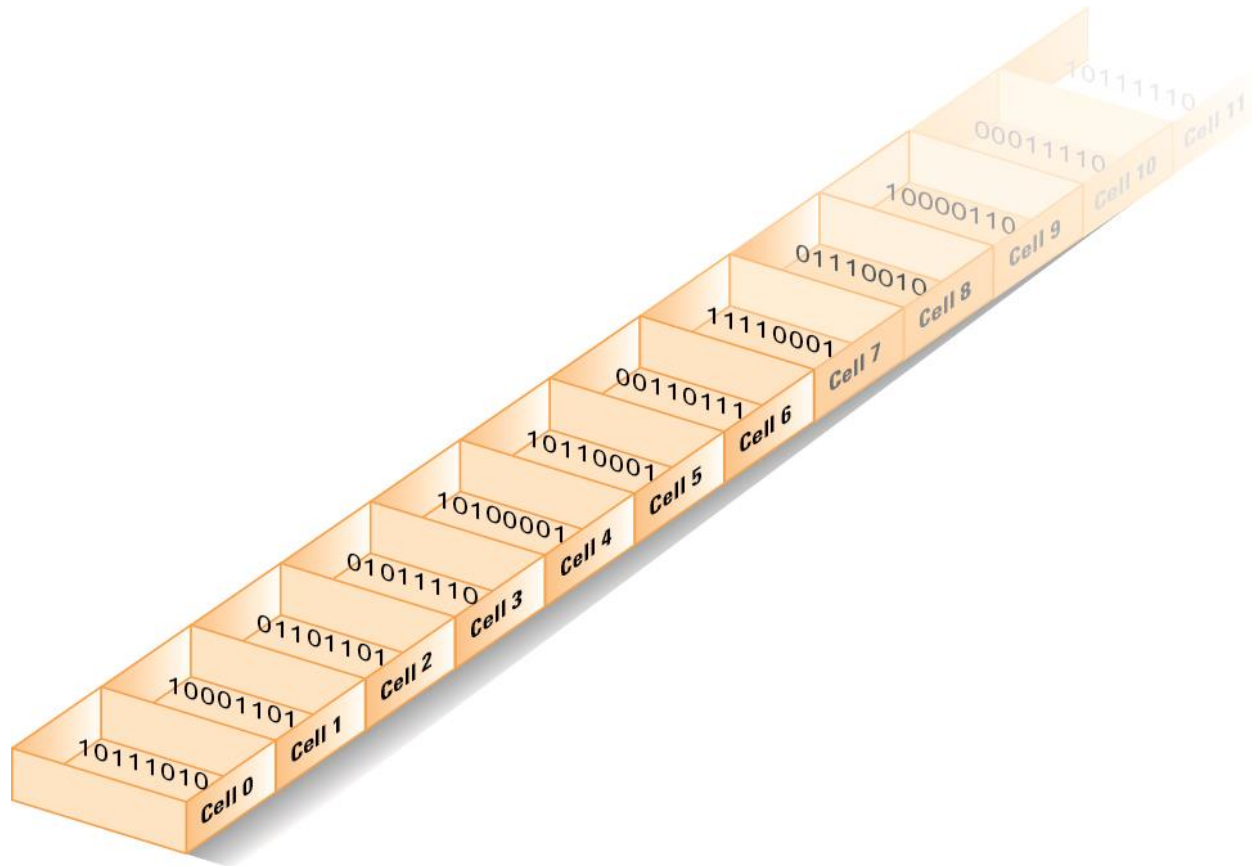
# Şekil 1.7 Bir byte(bayt) boyutundaki bellek hücresinin düzeni



# Ana Bellek Adresleri

- **Adres:** Bilgisayarın ana belleğindeki bir hücreyi eşsiz bir şekilde belirleyen bir 'isim'dir
  - Bu isimler aslında sayılardır.
  - Bu sayılar sıfırdan başlayarak ardışık olarak atanmış sayılardır.
  - Hücreleri sıralama tarzı, bellek hücreleriyle ilişkili bir şekildedir.

# Şekil 1.8 Adreslere göre düzenlenmiş bellek hücreleri



# Bellek Terimleri

- **Rastgele Erişimli Bellek(RAM):** Her türlü emirle kolayca erişilebilen, bağımsız hücrelerden oluşan bellektir.
- **Dinamik Bellek:** Geçici hafızadan oluşan RAM

# Bellek Kapasitesini Ölçme

- **Kilobyte:**  $2^{10}$  byte = 1024 byte
  - Örnek: 3 KB = 3 tane 1024 byte
- **Megabyte:**  $2^{20}$  byte = 1,048,576 byte
  - Örnek : 3 MB = 3 tane 1,048,576 byte
- **Gigabyte:**  $2^{30}$  byte = 1,073,741,824 byte
  - Örnek : 3 GB = 3 tane 1,073,741,824 byte

## 1.3 Yığın Depolama

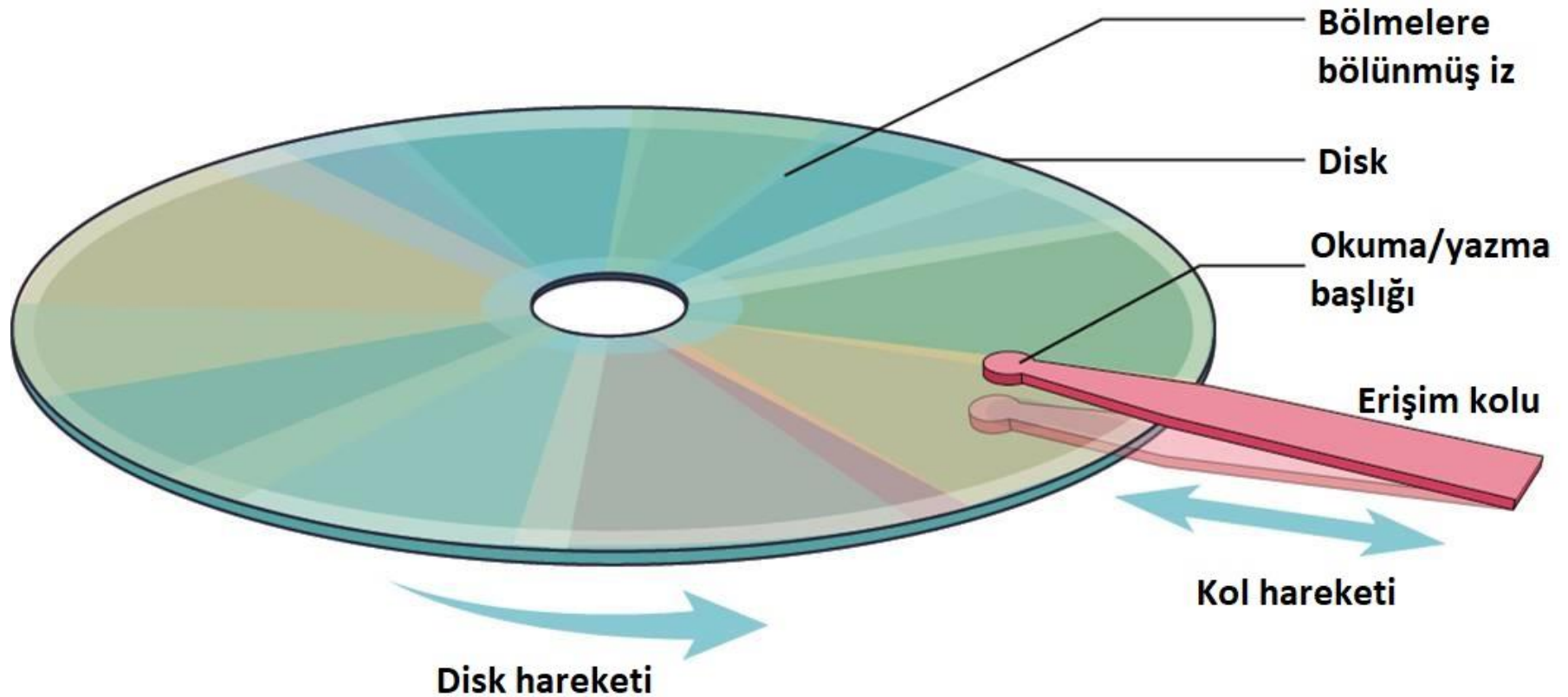
- Ek aygıtlar:
  - Manyetik diskler
  - Manyetik kasetler
  - CDler
  - Flaş sürücü
  - DVDler
  - Katı hal diskleri
- Ana belleğe sağladığı avantajlar
  - Kalıcılık
  - Daha geniş depolama kapasitesi
  - Düşük maliyet
  - Portatiflik(sök- tak yapılabilir)

# Yığın Depolama Performansı

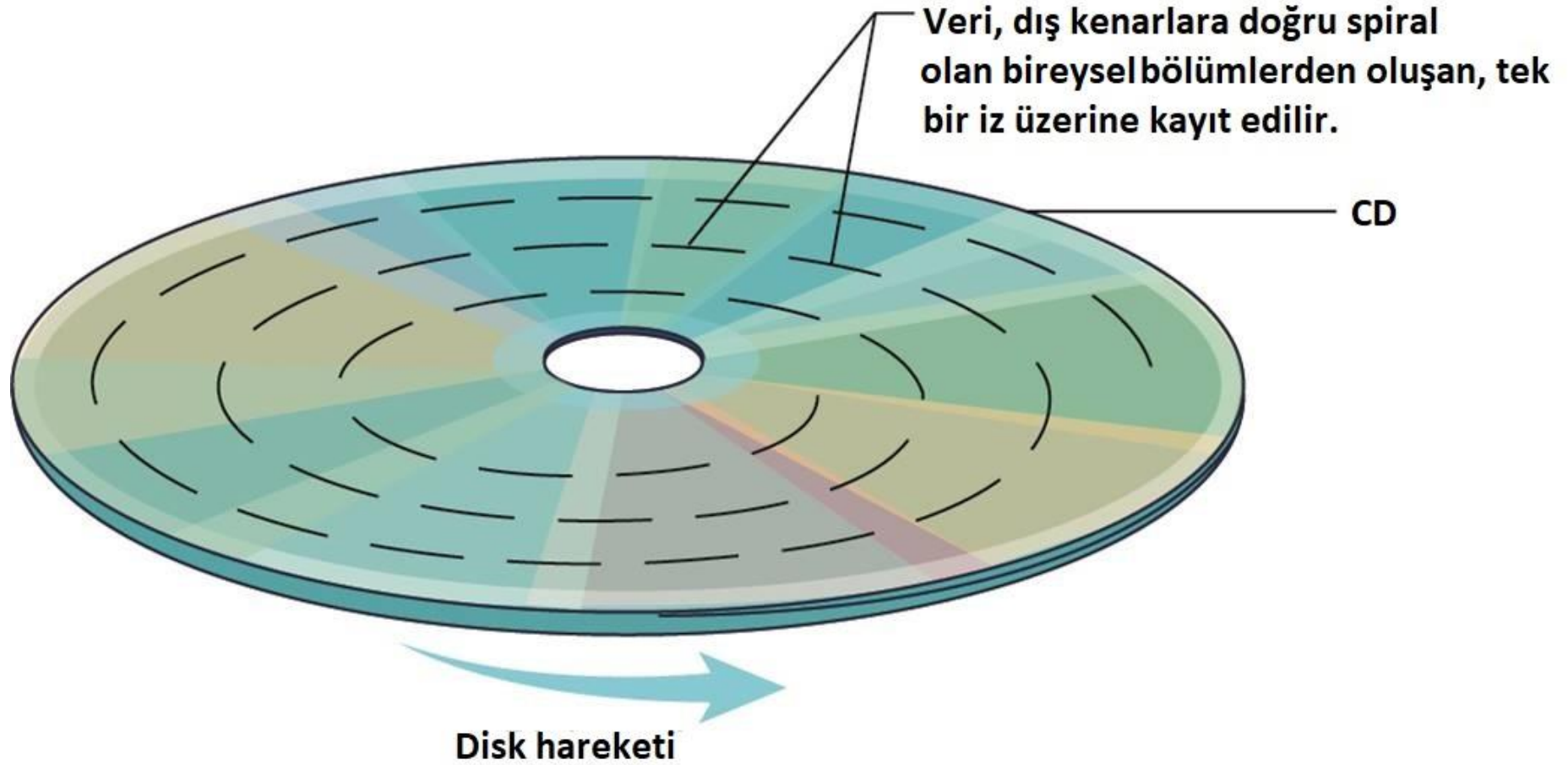
- **Bant genişliği:** Birim zamanda transfer edilebilen toplam bit miktarı
- **Gecikme:** Veri transferi isteğinin gönderilmesi ve verinin gelmesi arasında geçen zaman



# Şekil 1.9 Bir manyetik diskin depolama sistemi



## Şekil 1.10 CD depolama sistemi



# Flaş Sürücüler

- **Flaş bellek**– Elektronları minik silikon dioksit bölmelere hapseden devrelerdir.
- **Tekrar eden silme işlemi** zamanla içeriği yıpratır
- Yığın depolama genellikle şu cihazlarda kullanılır:
  - Dijital kameralar
  - Akıllı telefonlar
- **SD kartlar** depolama hacmini artırır.

# 1.4 Bilginin Bit Desenleri Cinsinden Gösterimi

- Bilginin bir çok türü bit desenleri şeklinde kodlanabilir
- Bilgiyi kodlama sistemi şunlara göre belirlenir:
  - Metin
  - Sayısal Veri
  - Görüntüler
  - Ses
  - Diğer veriler

# Metin Gösterimi

- Her karakter(harf, noktalama işareti, vb.) eşsiz bir bit desenine sahiptir.
  - **ASCII**: İngiliz alfabesindeki sembol ve ifadelerin gösterimi için 7-bit deseni kullanır.
  - **Unicode (UTF-8)**:Dünya genelinde kullanılan dillerdeki sembol ve ifadelerin gösterimi için 16-bit 21-bit arası desenleri kullanır

# Şekil 1.11

## ASCII veya UTF-8 kodlama şeklinde ‘Hello.’ mesajının gösterimi

01001000

**H**

01100101

**e**

01101100

**l**

01101100

**l**

01101111

**o**

00101110

**.**

# Sayısal Değerlerin Gösterimi

- **İkilik Sayı Sistemi:**  
Sayıları bitler şeklinde göstermek için 2 rakam kullanır
  - Bilgisayardaki bütün sayısal değerler 1ler ve 0lar dizisi şeklinde saklanır, depolanır.

Binary	Decimal	Hexadecimal
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F

# Görüntülerin Gösterimi

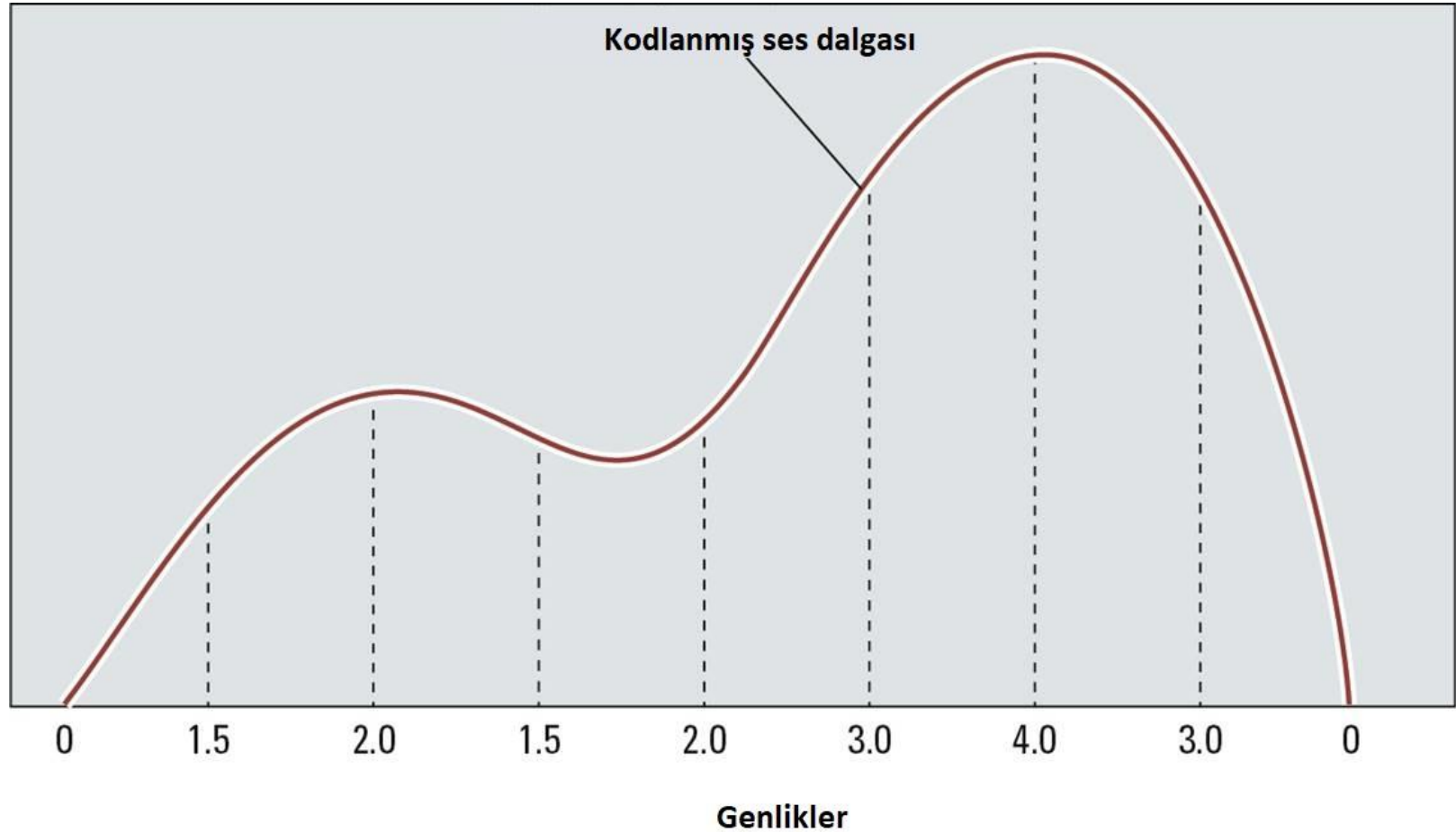
- Bit eşlem teknikleri
  - RGB: Kırmızı(Red), Yeşil(Green) ve Mavi(Blue) bileşenler
  - Piksel: Tek bir nokta gösterimini yapan ‘görüntü elemanı’
  - Parlaklık ve renklilik
  - Görüntü çoğaltma sorunları
- Vektör Teknikleri
  - Görüntüleri geometrik şekillerle gösterirler
  - Ölçeklenebilir
  - Truetype ve PostScript fontları birer örnektir



# Sesin Gösterimi

- Gerçek ses kaydını örnekleme teknikleri
  - Uzun mesafe telefonlar: 8000 örnek/saniye
  - CD sesi: 44,100 örnek/saniye

## Şekil 1.12 0, 1.5, 2.0, 1.5, 2.0, 3.0, 4.0, 3.0, 0 serisi tarafından gösterilen ses dalgası



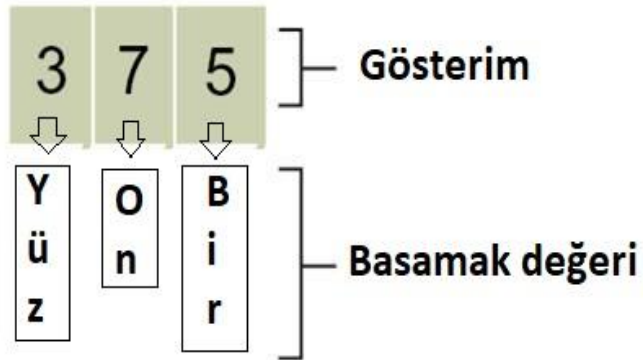
## 1.5 İkilik Sistem

Geleneksel ondalık sistem, onun kuvvetleri üzerine kurulmuştur.

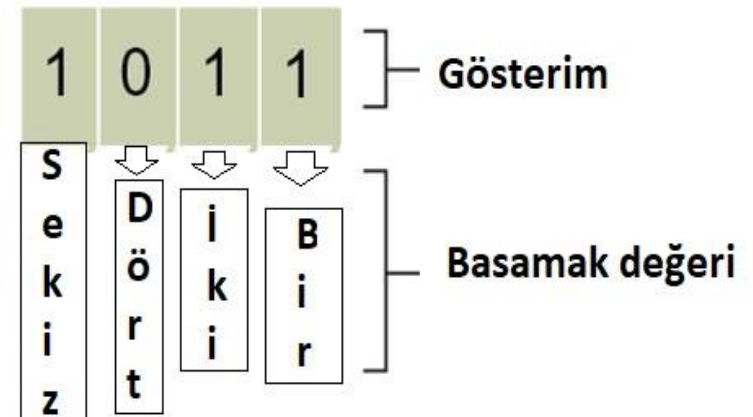
İkilik Sistem ise ikinin kuvvetleri baz alınarak oluşmuştur.

# Şekil 1.13 Onluk ve ikilik taban sistemleri

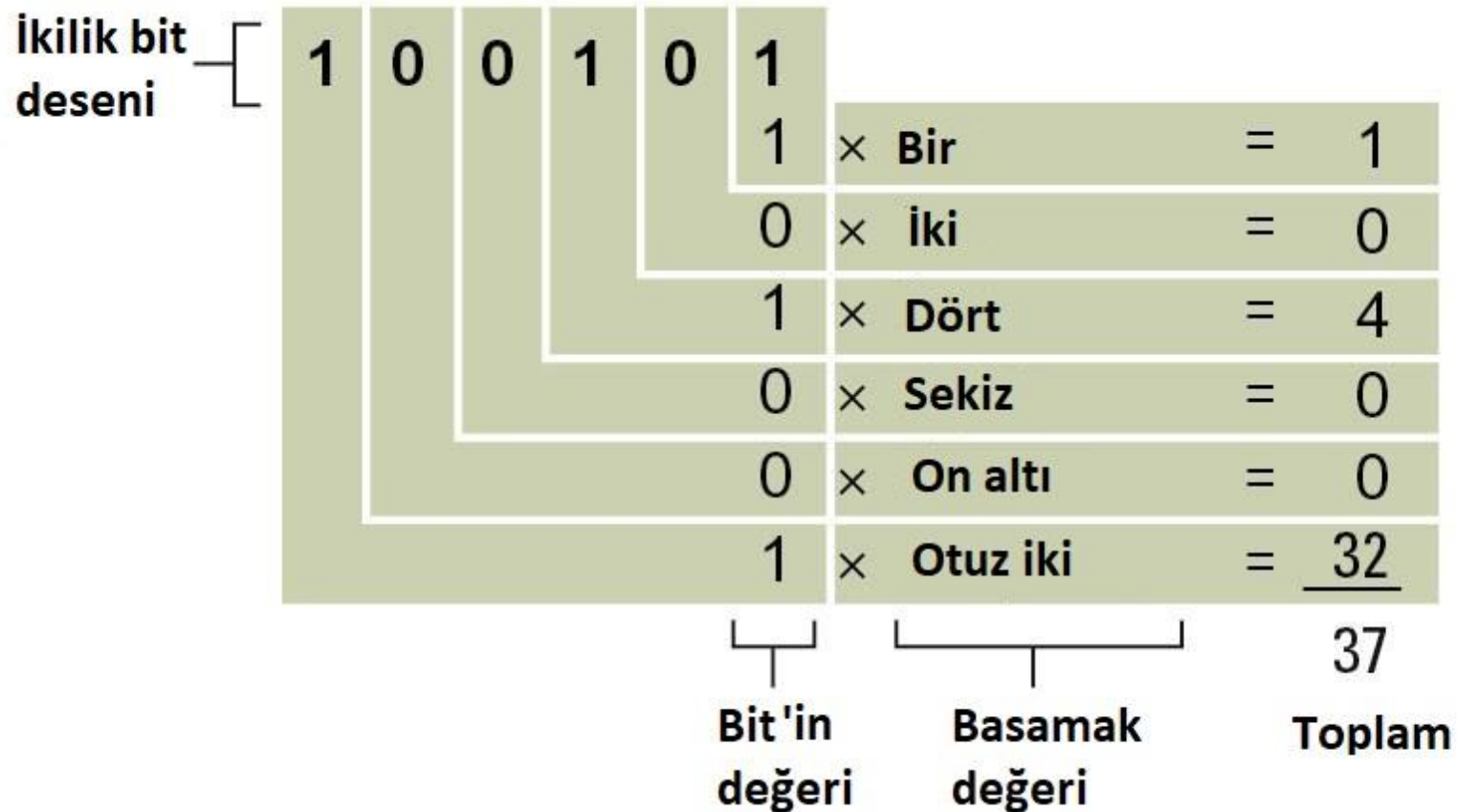
a. Onluk taban sistemi



b. İkilik taban sistemi



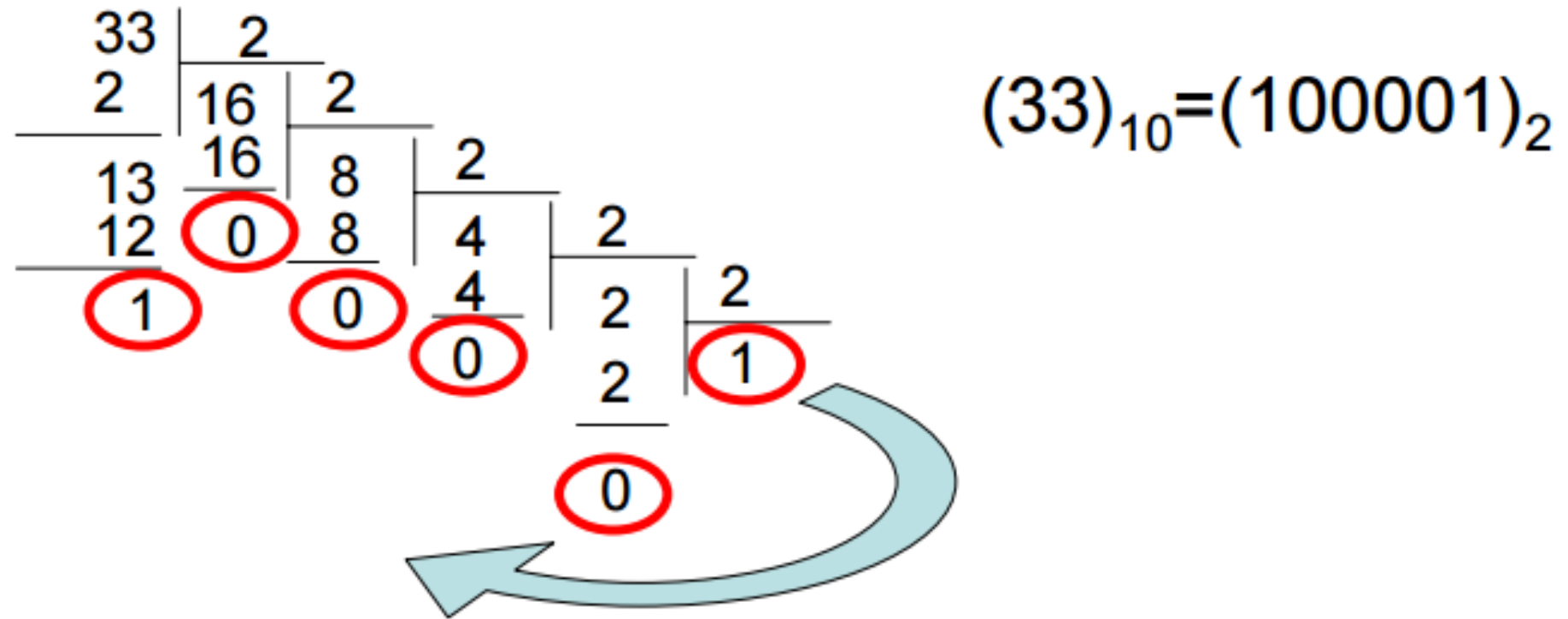
## Şekil 1.14 100101 ikilik gösterimini çözümleme



## Şekil 1.15 Pozitif bir tamsayının ikilik gösterimini bulmak için bir algoritma

- **Adım 1.** Değeri 2'ye böl ve kalan kısmını kaydet.
- **Adım 2.** Bölüm 0'a ulaşana kadar yeni bölümü 2'ye bölmeye devam et ve kalan kısmını kaydet.
- **Adım 3.** Bölüm değeri 0'a ulaştığı zaman, başlangıçtaki değerın ikilik gösterimi, kaydettiğimiz kalan değerlerinin sağdan sola doğru sıralanmış şeklidir

## Şekil 1.16 Otuz üç değerinin ikilik tabanda gösterimini bulmak için Şekil 1.15'teki algoritmanın uygulanması



## Şekil 1.17 İkilik sistemde toplama

$$\begin{array}{r} 0 \\ + 0 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 1 \\ + 0 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 0 \\ + 1 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 1 \\ + 1 \\ \hline 10 \end{array}$$

there are 10 types of people in this world,  
those who understand binary and those who don't



# Şekil 1.18 101.101 ikilik gösteriminin çözümlemesi

Bit deseni	1	0	1	.	1	0	1	
							1	× Sekizde bir = $\frac{1}{8}$
							0	× Dörtte bir = 0
							1	× İkide bir = $\frac{1}{2}$
							1	× Bir = 1
							0	× İki = 0
							1	× Dört = 4
								<u>5<math>\frac{5}{8}</math></u>
							Bitin değeri	Basamak değeri
								Toplam

## 1.6 Tam sayıların depolanması

- **İkinin tümleyeni gösterimi:** Tam sayıların en popüler gösterimidir.
- **Fazlalık (Excess) gösterimi:** Tam sayıların diğer bir gösterim şeklidir.

# Şekil 1.19 İkinin tümleyeni gösterimi

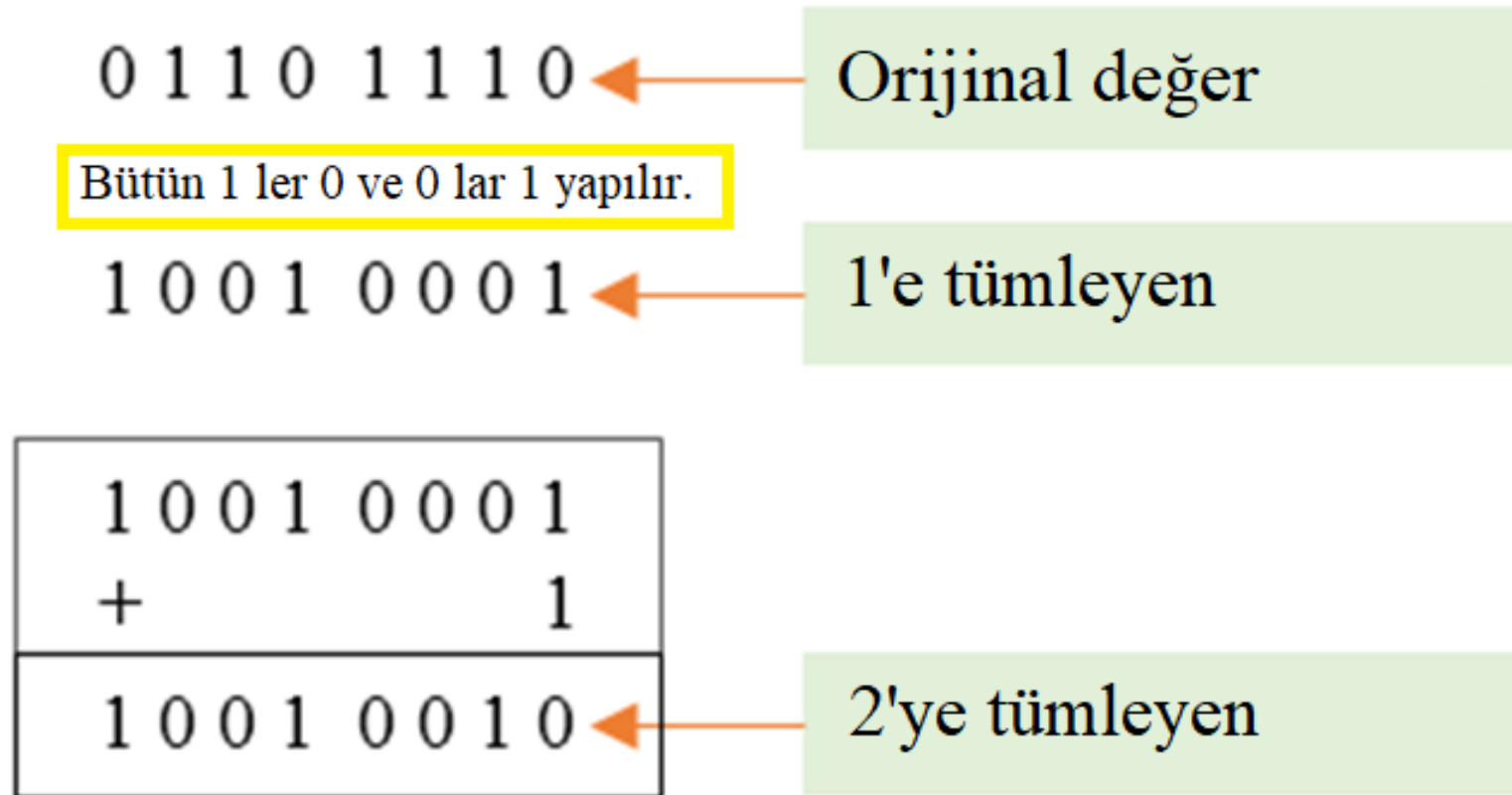
a.Üç bit uzunluğundaki desenlerde gösterim

Bit deseni	Gösterdiği değer
011	3
010	2
001	1
000	0
111	-1
110	-2
101	-3
100	-4

b.Dört bit uzunluğundaki desenlerde gösterim

Bit deseni	Gösterdiği değer
0111	7
0110	6
0101	5
0100	4
0011	3
0010	2
0001	1
0000	0
1111	-1
1110	-2
1101	-3
1100	-4
1011	-5
1010	-6
1001	-7
1000	-8

## Şekil 1.20 Dört bit kullanarak -6 değerini ikinin tümleyeni gösterimiyle kodlama



## Şekil 1.21 İkinin tümleyeniyle ilgili alıştırmalar

On tabanında işlem		İkinin tümleyeninde işlem		10 tabanında sonuç
$\begin{array}{r} 3 \\ + 2 \\ \hline \end{array}$	→	$\begin{array}{r} 0011 \\ + 0010 \\ \hline 0101 \end{array}$	→	5
$\begin{array}{r} -3 \\ + -2 \\ \hline \end{array}$	→	$\begin{array}{r} 1101 \\ + 1110 \\ \hline 1011 \end{array}$	→	-5
$\begin{array}{r} 7 \\ + -5 \\ \hline \end{array}$	→	$\begin{array}{r} 0111 \\ + 1011 \\ \hline 0010 \end{array}$	→	2

# Taşma Problemi (Overflow)

- Her sistemin gösterimini yapabileceği değerlerin bir sınırı vardır
- **Taşma(Overflow)**
  - Bir hesaplama işlemi, makinede gösterilebilecek değerlerin sınırları dışında bir değer ürettiğinde meydana gelir
  - Eğer sonuç işaret biti yanlışsa, bir taşma gerçekleşir
  - 16 bit sistemler 32 bit sistemlere yükseltilmiştir

# Şekil 1.22 Fazlalık sekiz (excess eight) dönüşüm tablosu

Bit deseni	Gösterdiği değer
1111	7
1110	6
1101	5
1100	4
1011	3
1010	2
1001	1
1000	0
0111	-1
0110	-2
0101	-3
0100	-4
0011	-5
0010	-6
0001	-7
0000	-8

## Şekil 1.23 Üç bit uzunluğundaki desenleri kullanan «fazlalık sekiz» gösterim sistemi

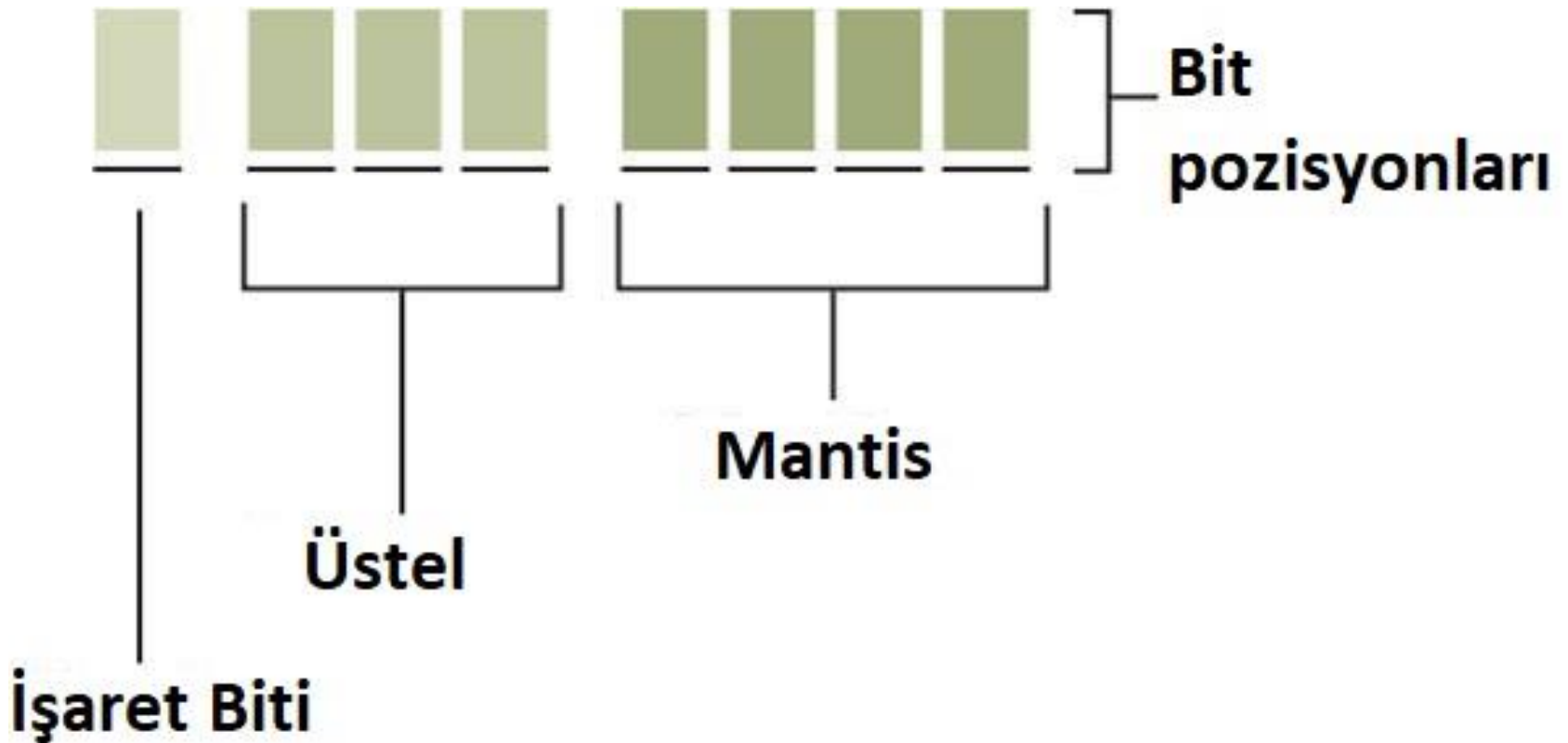
Bit deseni	Gösterdiği değer
111	3
110	2
101	1
100	0
011	-1
010	-2
001	-3
000	-4



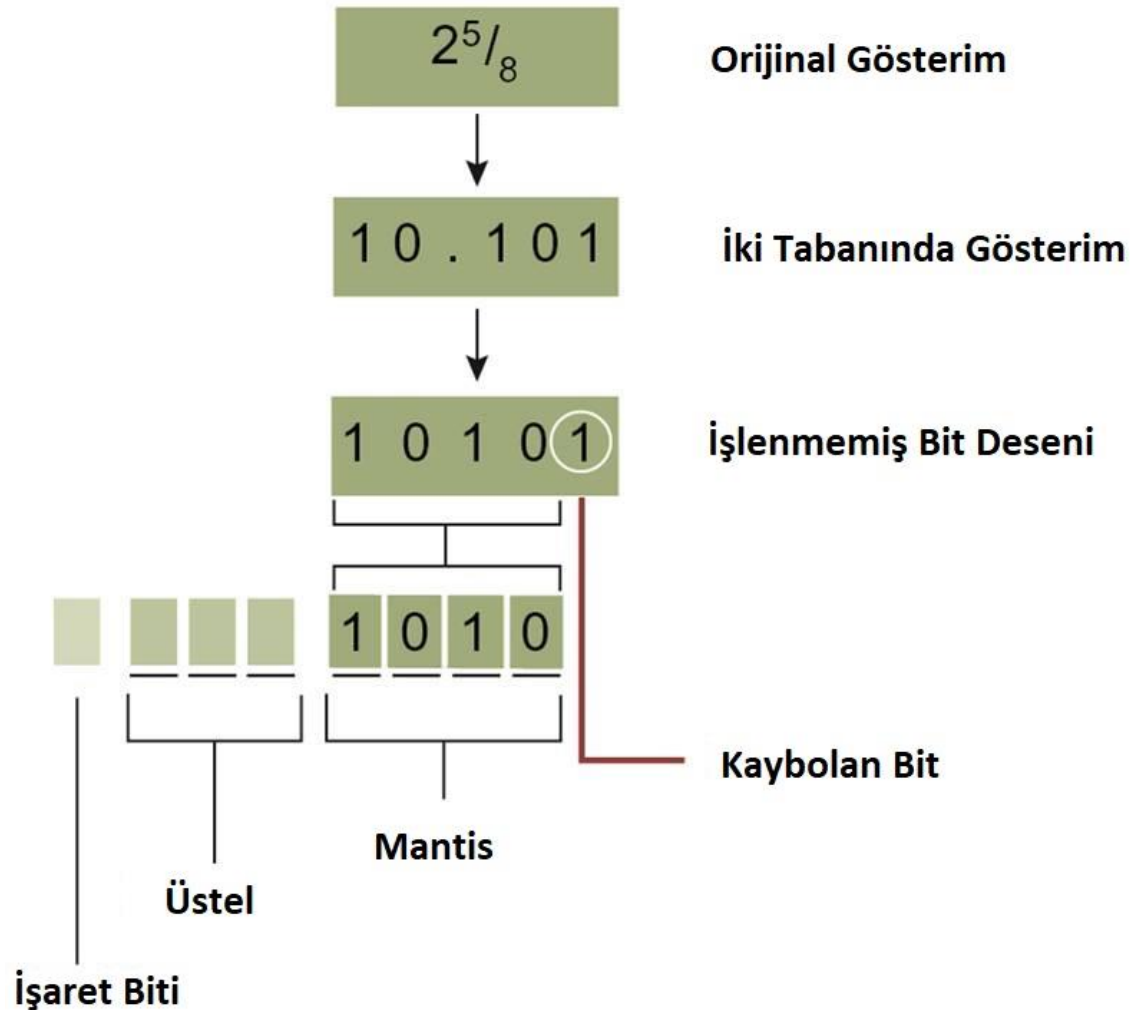
## 1.7 Kesirleri Depolama

- **Kayar nokta (floating point) gösterimi:** İşaret biti, üstel alan ve mantis alanından oluşur.
  - **Normalize form:** En soldaki 1 değerinden başlayarak mantis alanını doldur

# Şekil 1.24 Kayan nokta gösterimi elemanları



# Şekil 1.25 $2^{5/8}$ değerinin çözümlenmesi



# Kırpma (Yuvarlama) Hataları

- Mantis yeterince büyük olmadığı için depolanan verinin bir kısmının kaybolması durumunda ortaya çıkar
- Sonsuz kesir açılımları
  - Daha çok ikilik sayı gösteriminde görülür
  - Bu değerler genellikle tamsayılara dönüştürülür

## 1.8 Veri ve Programlama

Bir *programlama dili*, insanların yüksek seviyede soyutlama kullanarak bilgisayarlara algoritmaları tam olarak ifade etmesini sağlamak için oluşturulmuş bilgisayar sistemleridir.

# Python ile Çalışmaya Başlama

- **Python:** Uygulamalar ve bilimsel hesaplama için kullanılan ve öğrenciler için dili tanıtıcı görev üstlenen popüler bir programlama dilidir.
- [www.python.org](http://www.python.org)\_adresinden ücretsiz bir biçimde indirilebilir.
- Python yorumlanabilir bir dildir
  - Yazım:

```
print('Hello, World!')
```

- Çıktı:

```
Hello, World!
```

# Değişkenler

- **Değişkenler:** Sonra kullanmak için veri adlandırma
- Cebirdeki matematik değişkenlerine benzerdir

```
s = 'Hello, World!'
print(s)
```

```
my_integer = 5
my_floating_point = 26.2
my_Boolean = True
my_string = 'characters'
my_integer = 0xFF
```

# Operatörler ve İfadeleri

```
print(3 + 4)      # 7 Çıktısı verir  
print(5 - 6)      # -1 Çıktısı verir  
print(7 * 8)      # 56 Çıktısı verir  
print(45 / 4)     # 11.25 Çıktısı verir  
print(2 ** 10)    # 1024 Çıktısı verir
```

```
s = 'hello' + 'world'  
s = s * 4  
print(s)
```



# Para Birimi Dönüştürme

```
# Para birimi dönüştürme için bir program.

USD_to_GBP = 0.66      # Bugünün döviz kuru
GBP_sign = '\u00A3'    # £ sembolü için unicode
değeri

dollars = 1000          # Çevrilecek dolar miktarı

# Çevrim hesaplamaları
pounds = dollars * USD_to_GBP

# Sonucu yazdırma
print('Today, $' + str(dollars))
print('converts to ' + GBP_sign + str(pounds))
```

# Debugging

- *Sözdizim (Syntax) hataları*

```
print(5 +)
```

SyntaxError: invalid syntax

```
pront(5)
```

NameError: name 'pront' is not defined

- *Anlamsal Hatalar (Semantic errors)*

- Hatalı ifadeler

```
total_pay = 40 + extra_hours * pay_rate
```

- *Çalışma Zamanı Hataları (Runtime errors)*

- Sıfıra bölüm işlemine rastlama

# 1.9 Veri Sıkıştırma

- Kayıplı ve Kayıpsız
- Kodlama
  - İşlem uzunluğu kodlama
  - Frekans-bağımlı kodlama  
(Huffman kodları)
  - Göreceli kodlama
  - Sözlük kodlama

# Görüntüleri Sıkıştırma

- GIF: Çizgi filmler için uygundur
- JPEG: Fotoğraflar içindir
- TIFF: Görüntü arşivleme için idealdir

# Ses ve Video Sıkıştırma

- MPEG
  - Yüksek kalite televizyon yayını
  - Video konferans
- MP3
  - Ses
  - Müzik

## 1.10 İletişim Hataları

- Hedef: Hesaplama aygıtının güvenilirliğini artırıp hatayı en aza indirmek
- Eşlik (parity) bitleri

# Şekil 1.26 Tek eşlik için A ve F kodlarının ayarlanması

