

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 İkilik 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



GPT sor

Bileştiğin
her şey bir 1'ni
sıfır bire sıfırını
1 yapmaz

Program buna
nisi 1'ni 0
nisi 0'ları 1
yapmaz için
fordunur
olar

1.1 Bitler ve Depolanmaları

- **Bit:** İkilik Rakamlarıdır(0 veya 1)
- Bit desenleri bilgiyi ifade etmek için kullanılırlar
 - Sayılar
 - Harfler
 - Resimler (2 boyutlu bir matris)
 - Ses
 - Ve diğer veriler

tıpkı 1 ve 0 la ifade edilir

Boolean İşlemleri

- **Boolean İşlemi:** Bir veya daha fazla doğru/yanlış değerini yöneten işlemidir (1) (0)
- Özel işlemler
 - AND (Ve)
 - OR (Veya)
 - XOR (Ayrıcalıklı OR)
 - NOT (Değil)

(George Boole)
* Boole Cabri

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

AND(Ve) işlemi

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

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

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

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

* Bu işlenler
elekt. okul
yapar derslerde

“Matik
Dersleri
Derir”

OR(Veya) işlemi

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

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

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

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

XOR(Özel OR)

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

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

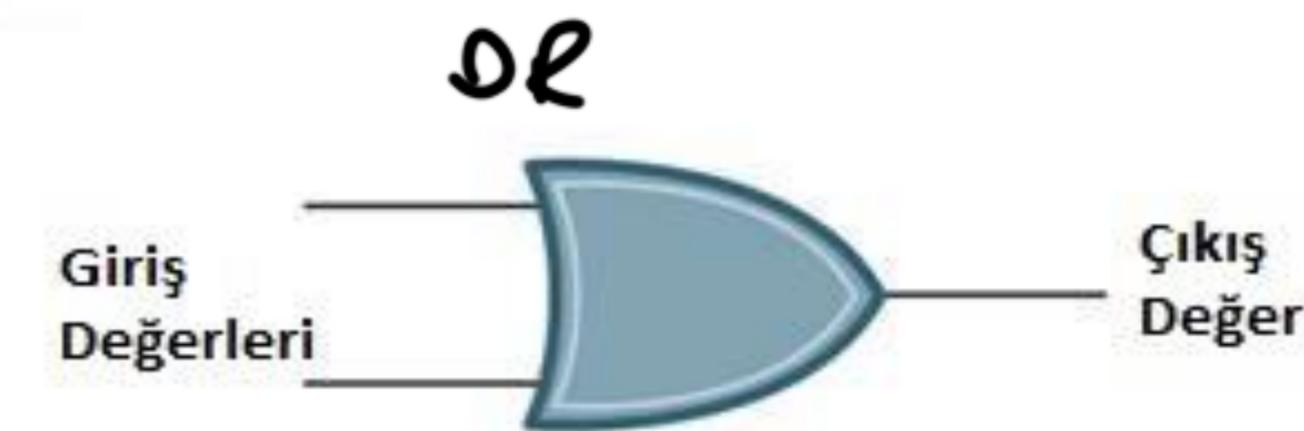
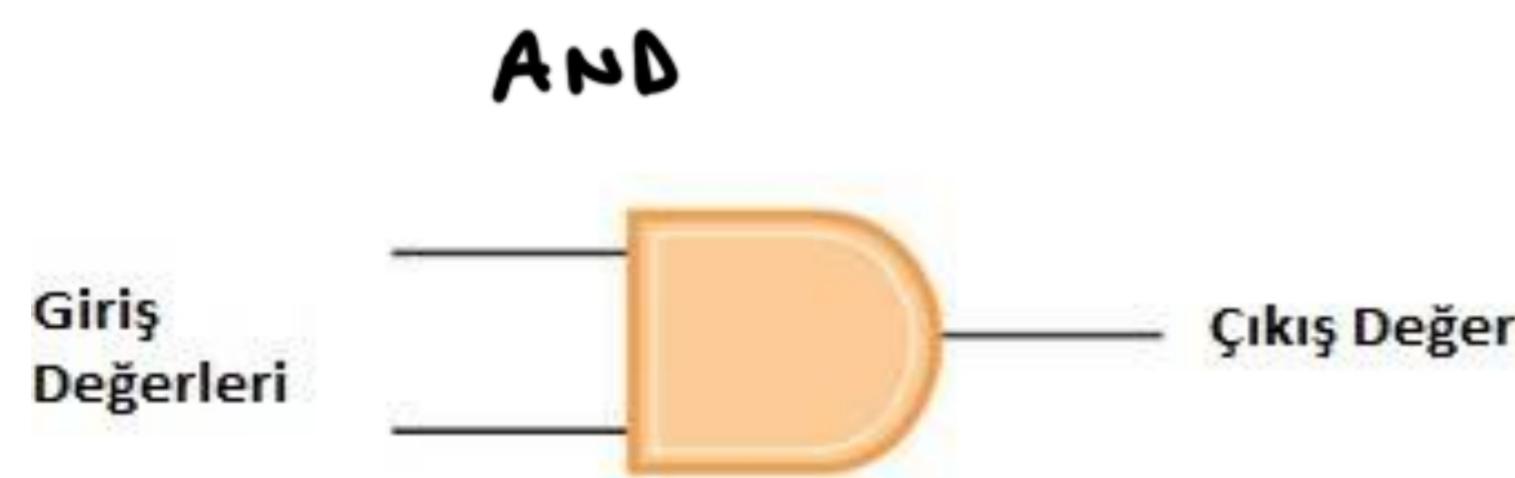
$$\begin{array}{r} 1 \\ \text{XOR} \\ \hline 1 \end{array}$$

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

Kapılar

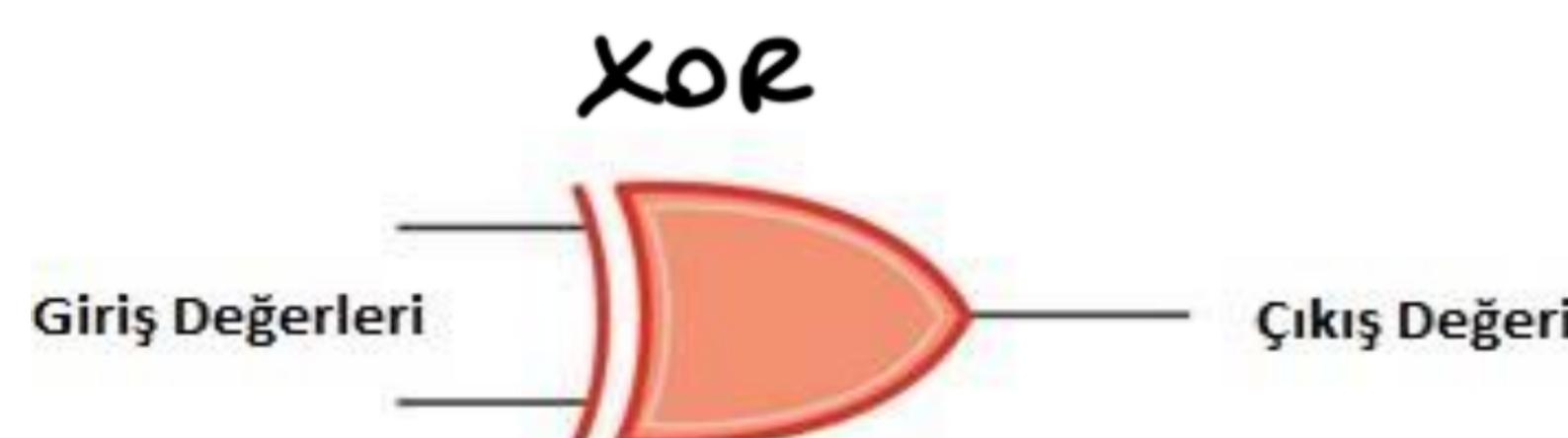
- **Kapı:** Boolean işlemini hesaplayan bir aygittır
 - Genelde transistör denilen küçük elektronik devreler şeklinde kullanılırlar *transistor icadı ile biliç yapıldı.*
 - Çeşitli teknolojilerle etkileşimde bulunabilir, inşa edilebilirler
 - Bilgisayarların oluşturduğu yapı bloklarını sağlamaktadırlar

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

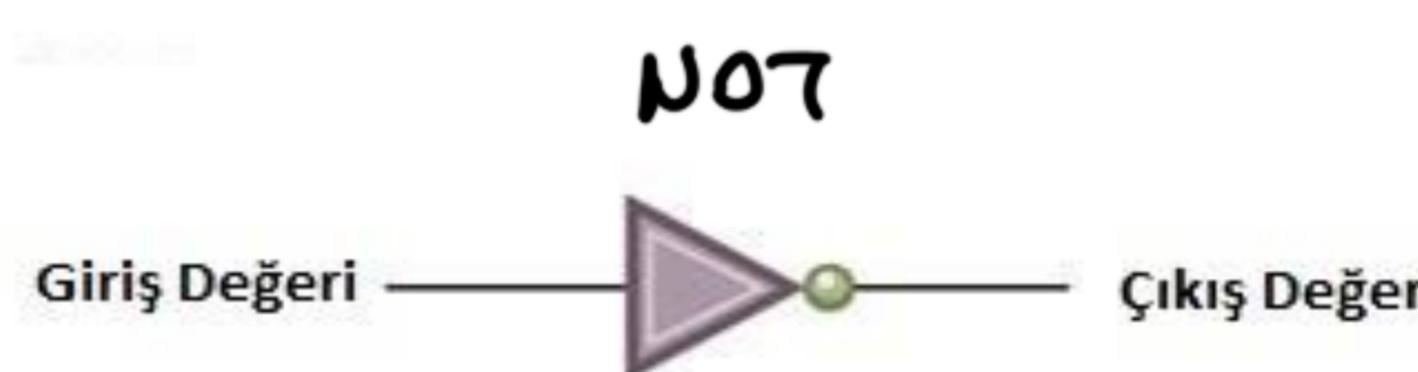


Giriş Değerleri	Çıkış Değeri
0 0	0
0 1	0
1 0	0
1 1	1

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



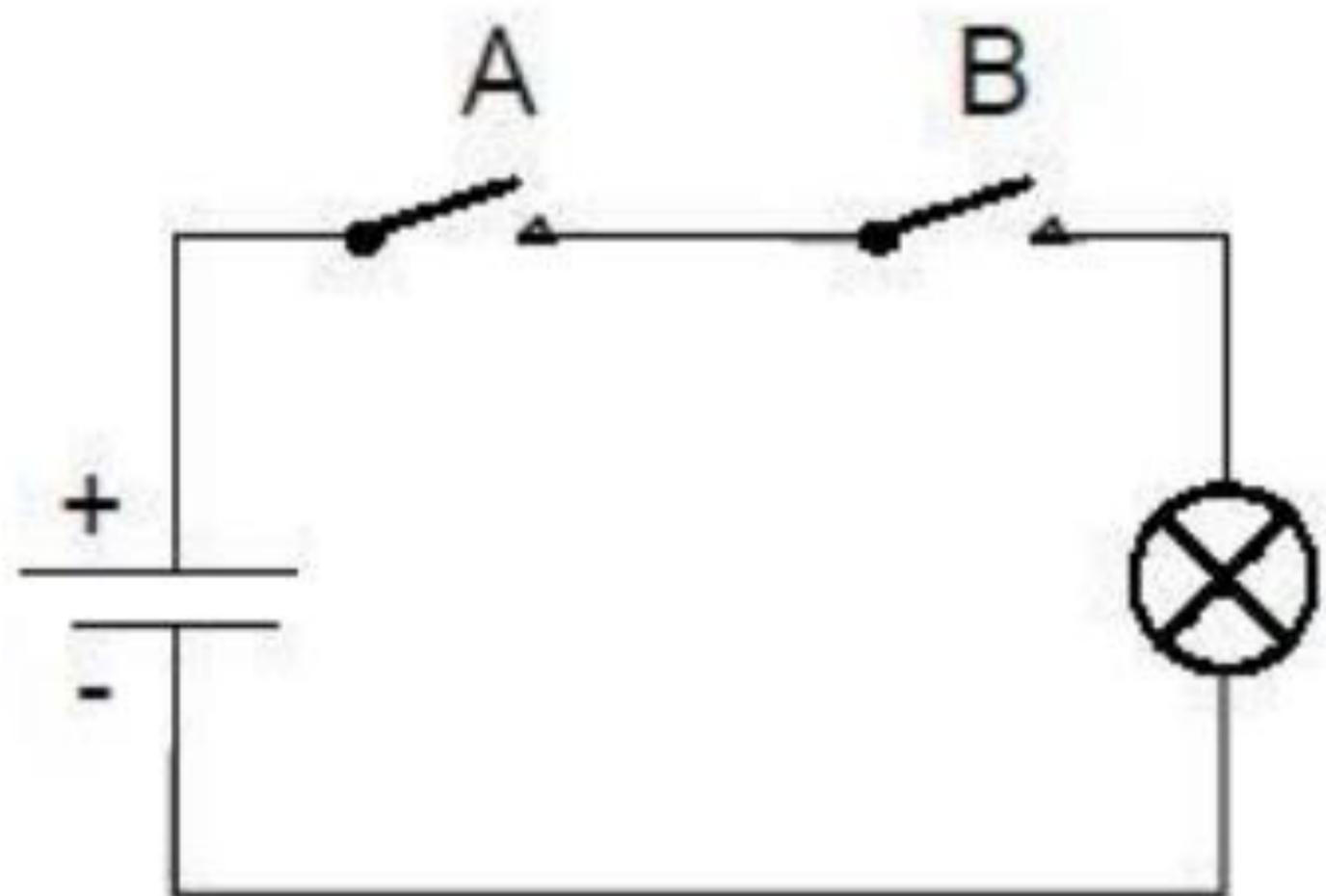
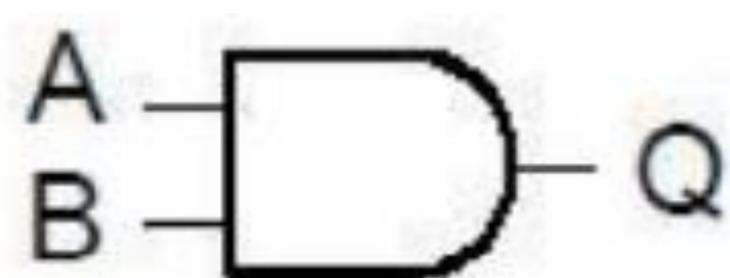
Giriş Değerleri	Çıkış Değeri
0 0	0
0 1	1
1 0	1
1 1	0



Giriş Değeri	Çıkış Değeri
0	1
1	0

$SV \rightarrow 1$ bilgisi
 $0, TV \rightarrow 0$ bilgisi

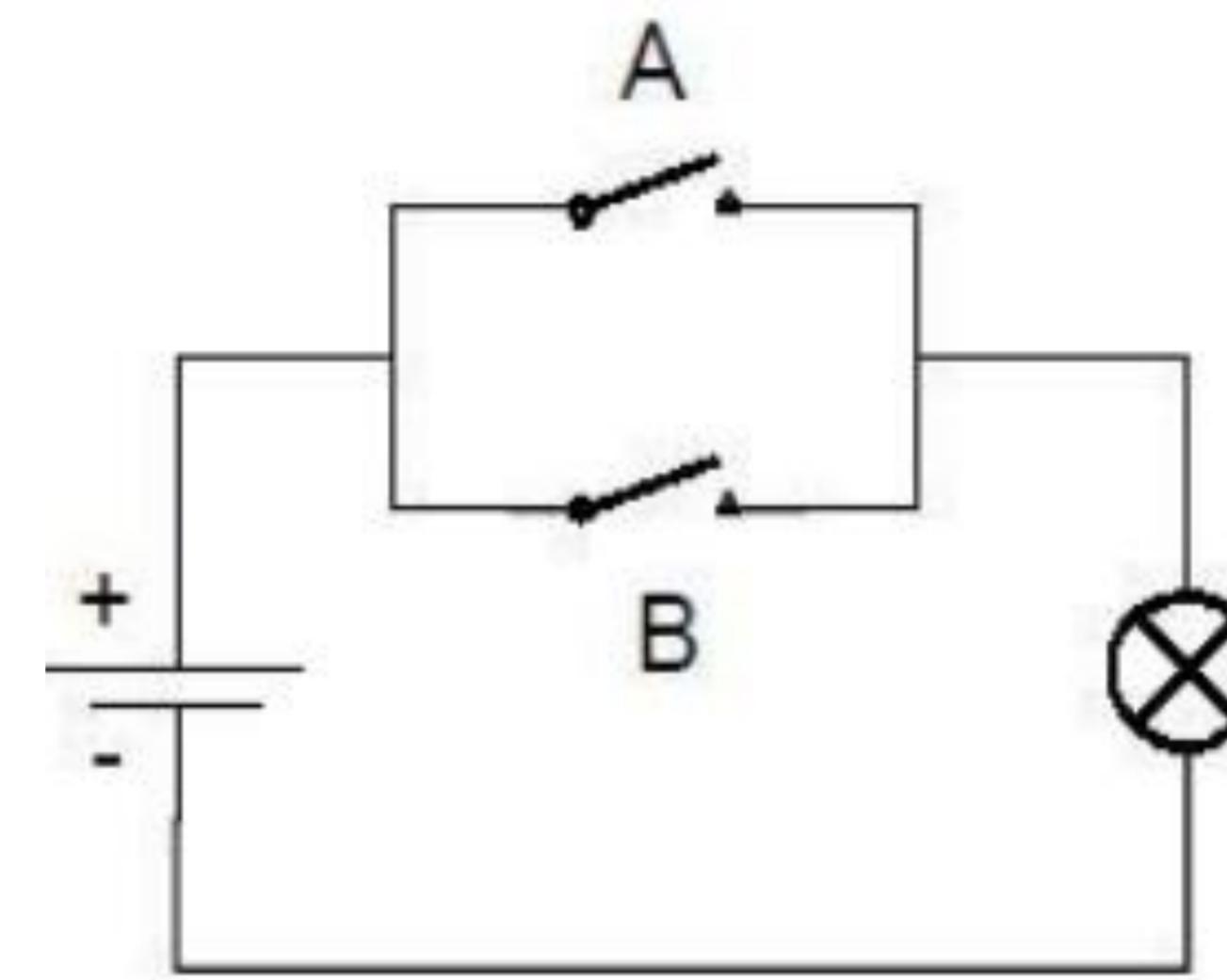
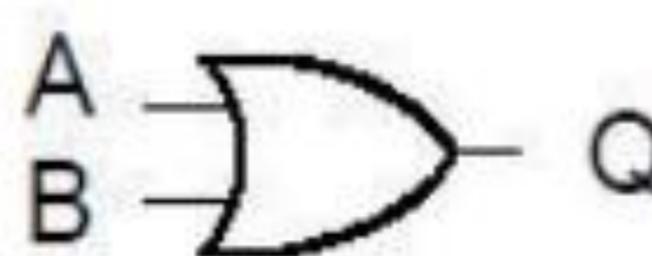
S  S çıkar



AND

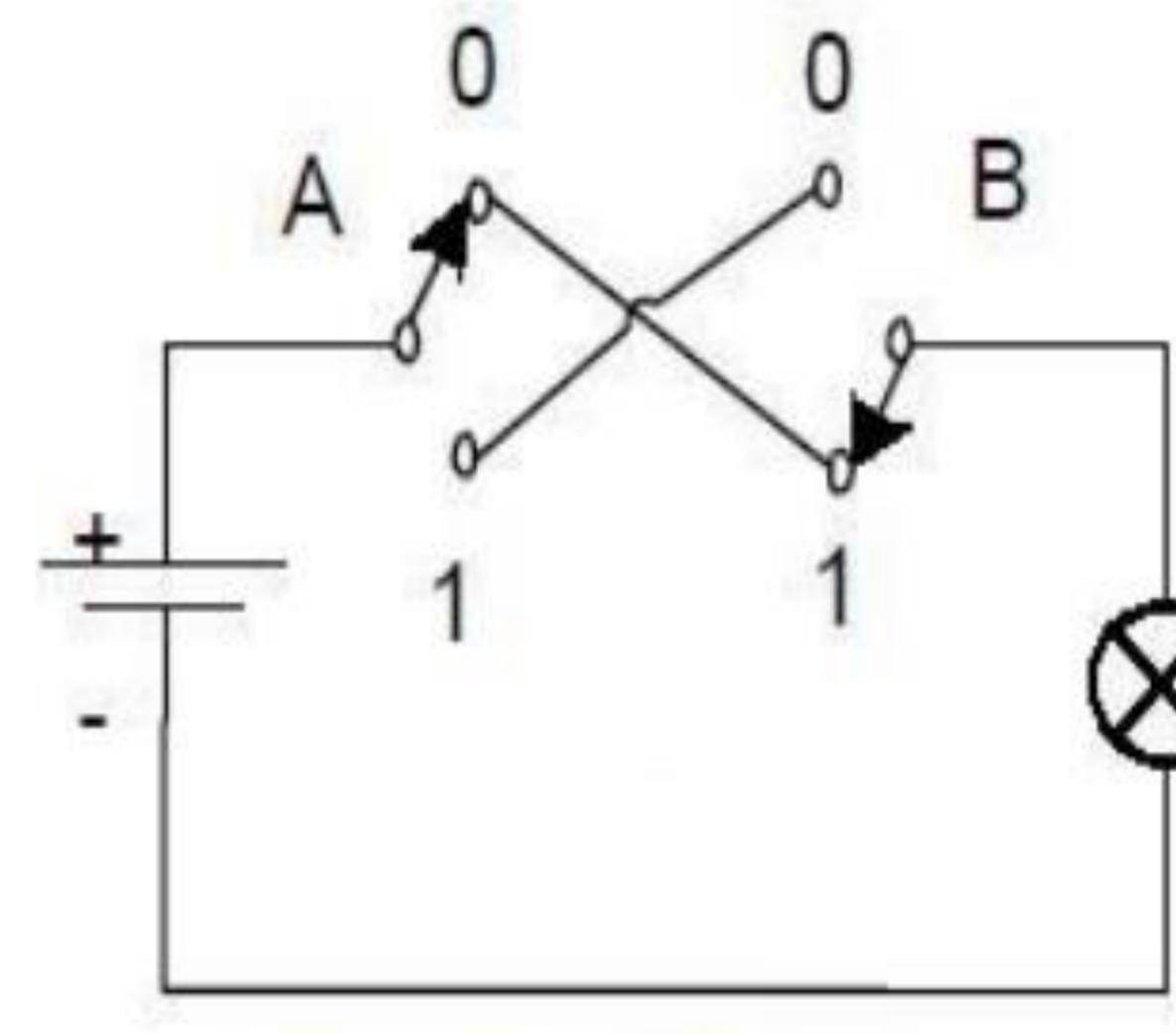
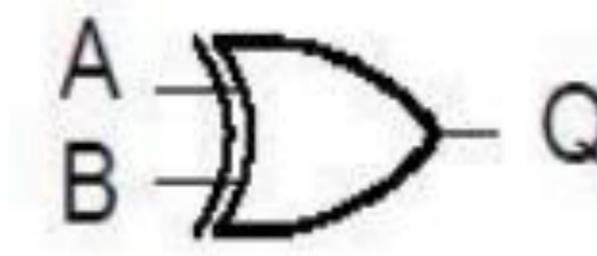
Sekildeki derreye
esdeger islem gorur.

* Temsil etmek icin



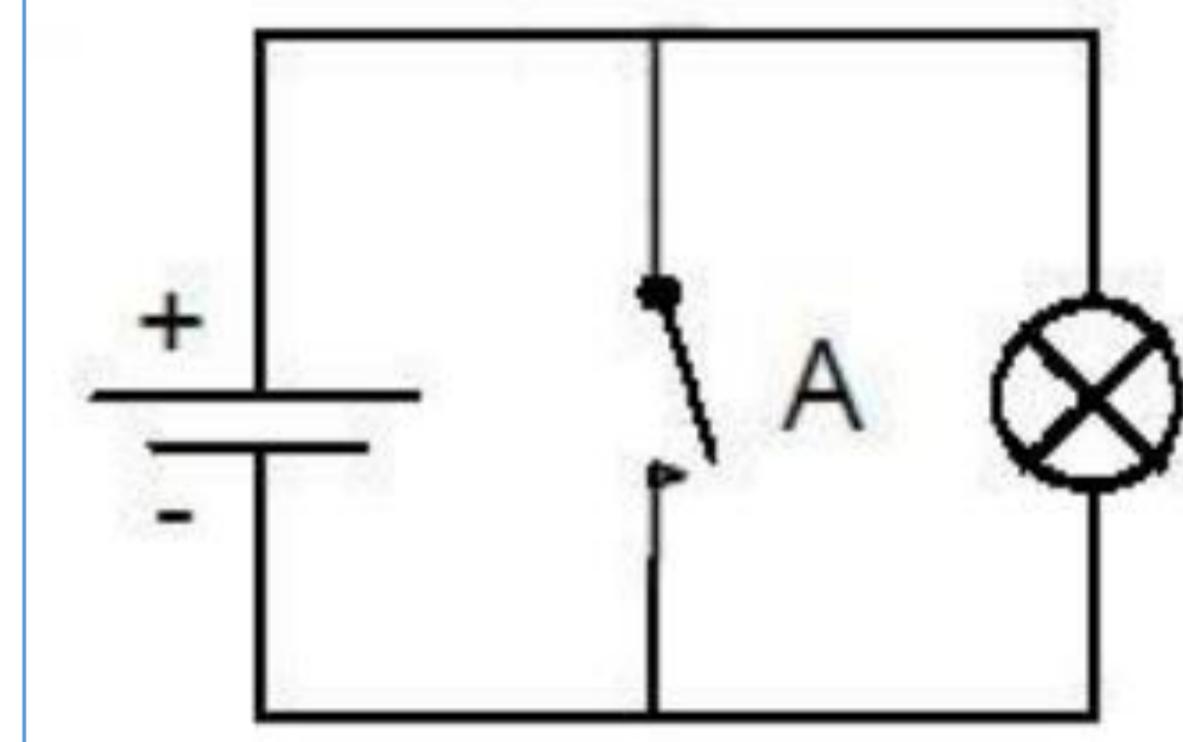
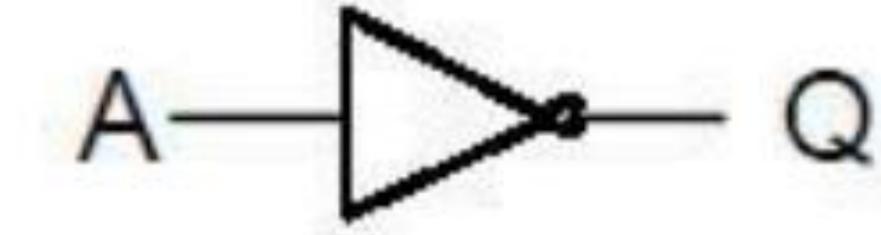
OR

* int. baki Transistor ^{öb} AND Gate



XOR

Mikro islemler
milyonlar icir.



NOT

Kisa devre
etraek

bunlarda

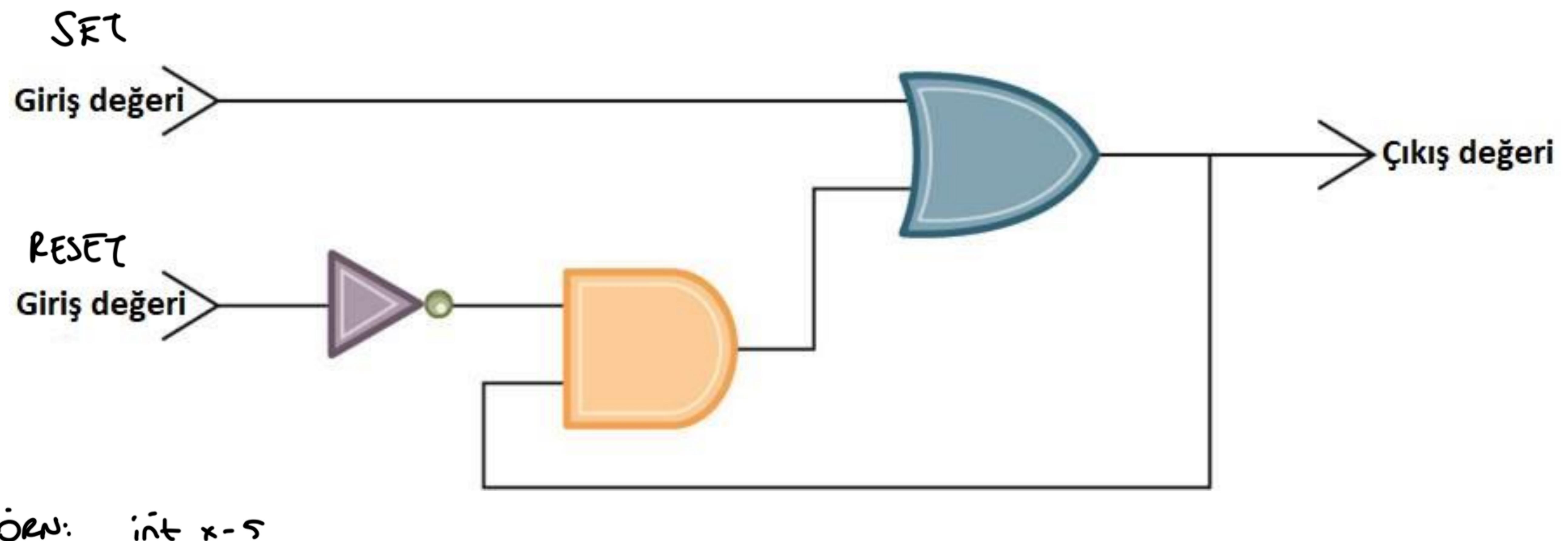
Flip-floplar

bölüm derleme

Mass Storage
Hard Disk : Depolama aygıtları RAM ile CPU arasında etkileşim olan sanal hard disk
'siz big. olabilir ama RAM'siz olmaz
Bilg. cihazları sistem işlevini RAM'de yapır.
RAM

- 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



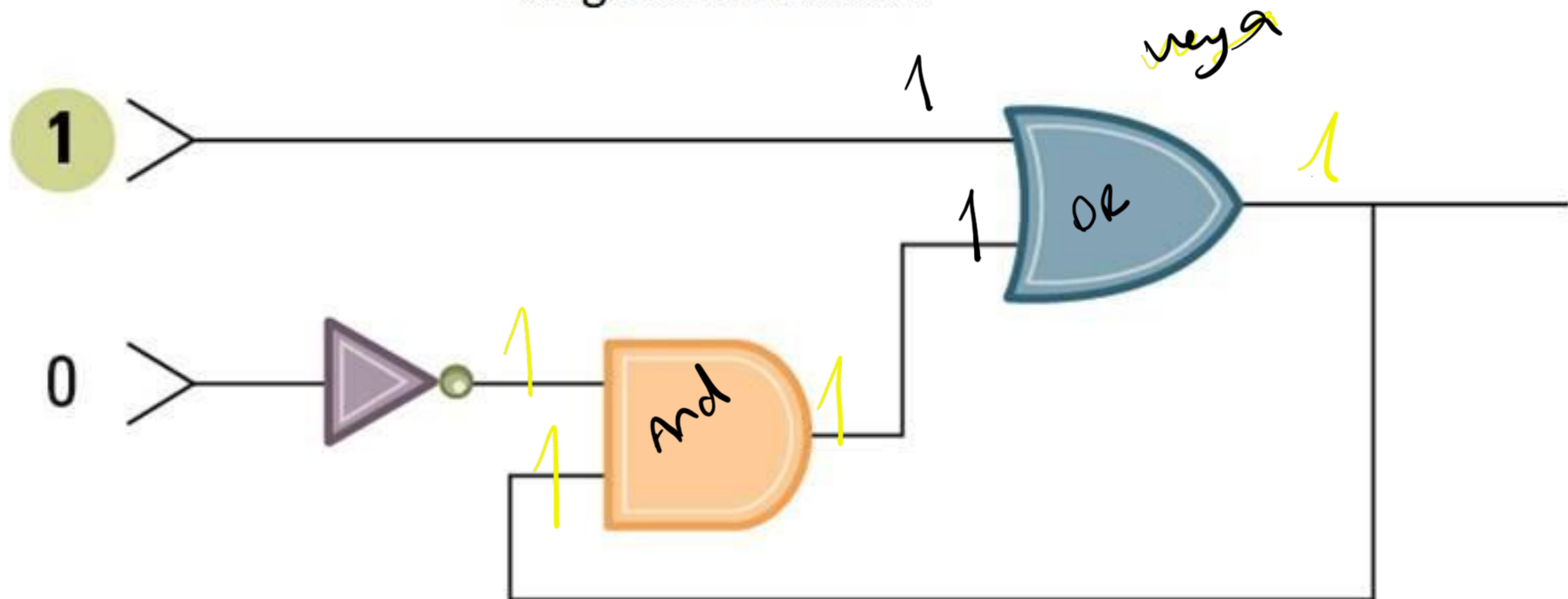
ÖRN: int x=5

x
cell0

A handwritten diagram showing the multiplication of two binary numbers, 101 and 101. The result is shown in a box with a handwritten 'x' above it and 'cell0' to its right.

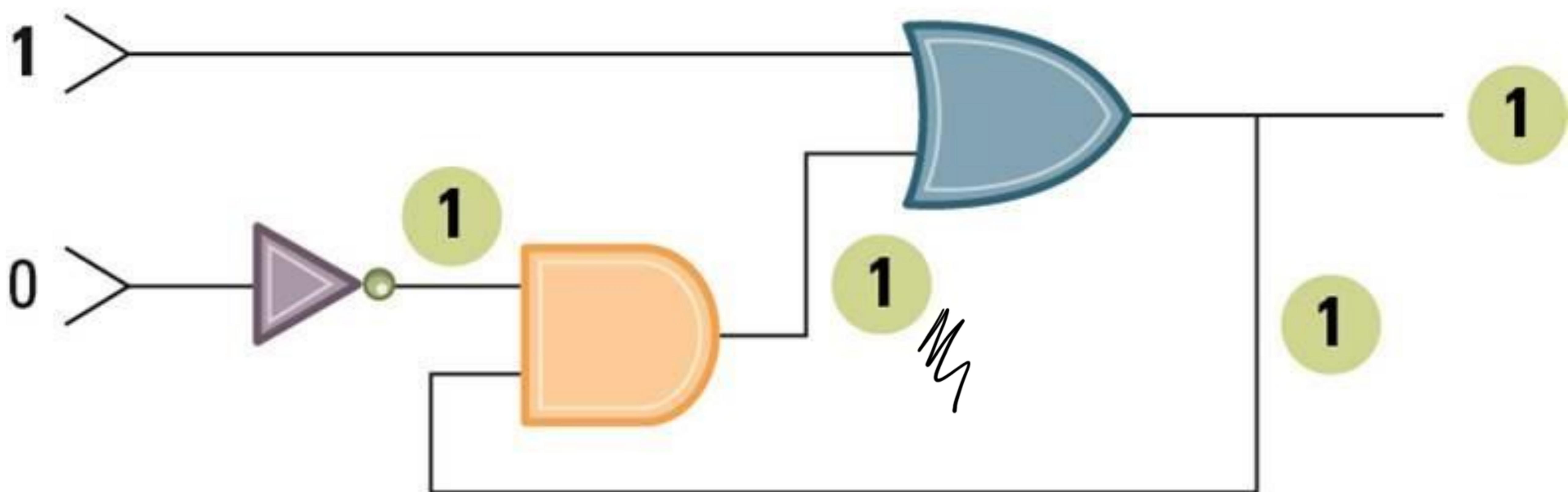
Şekil 1.4 Bir flip-flop'un çıkış değerini 1'e ayarlama

a. İlk olarak üstteki giriş değerine 1 atanır.



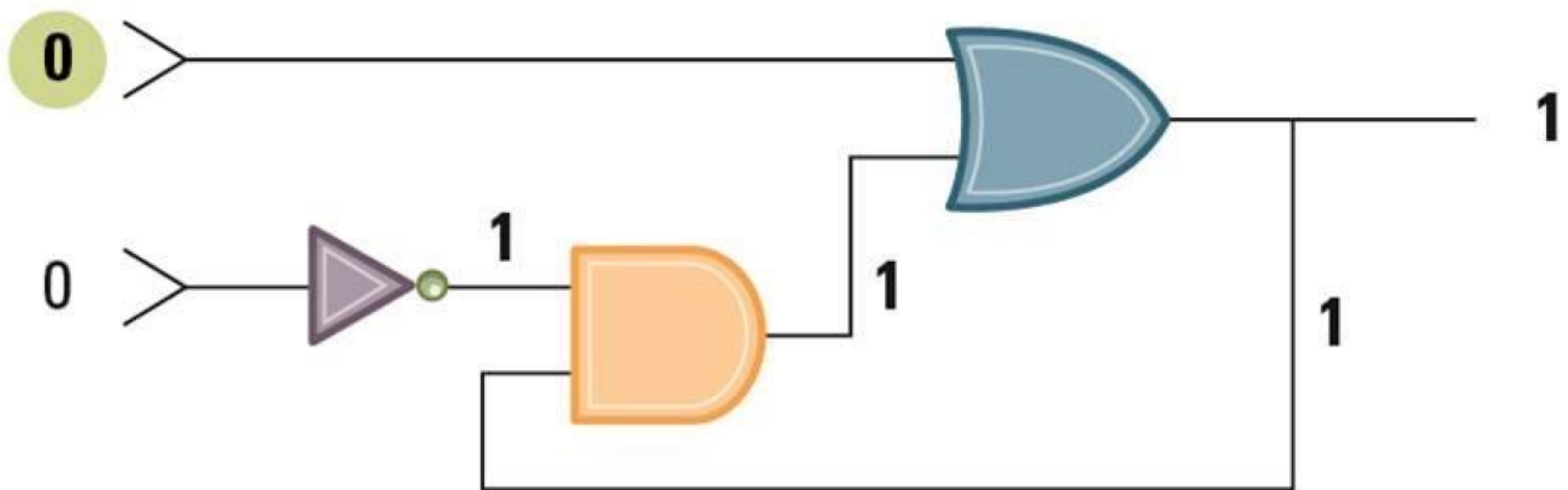
Şekil 1.4 Bir flip-flop'un çıkış değerini 1'e ayarlama (devamı)

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

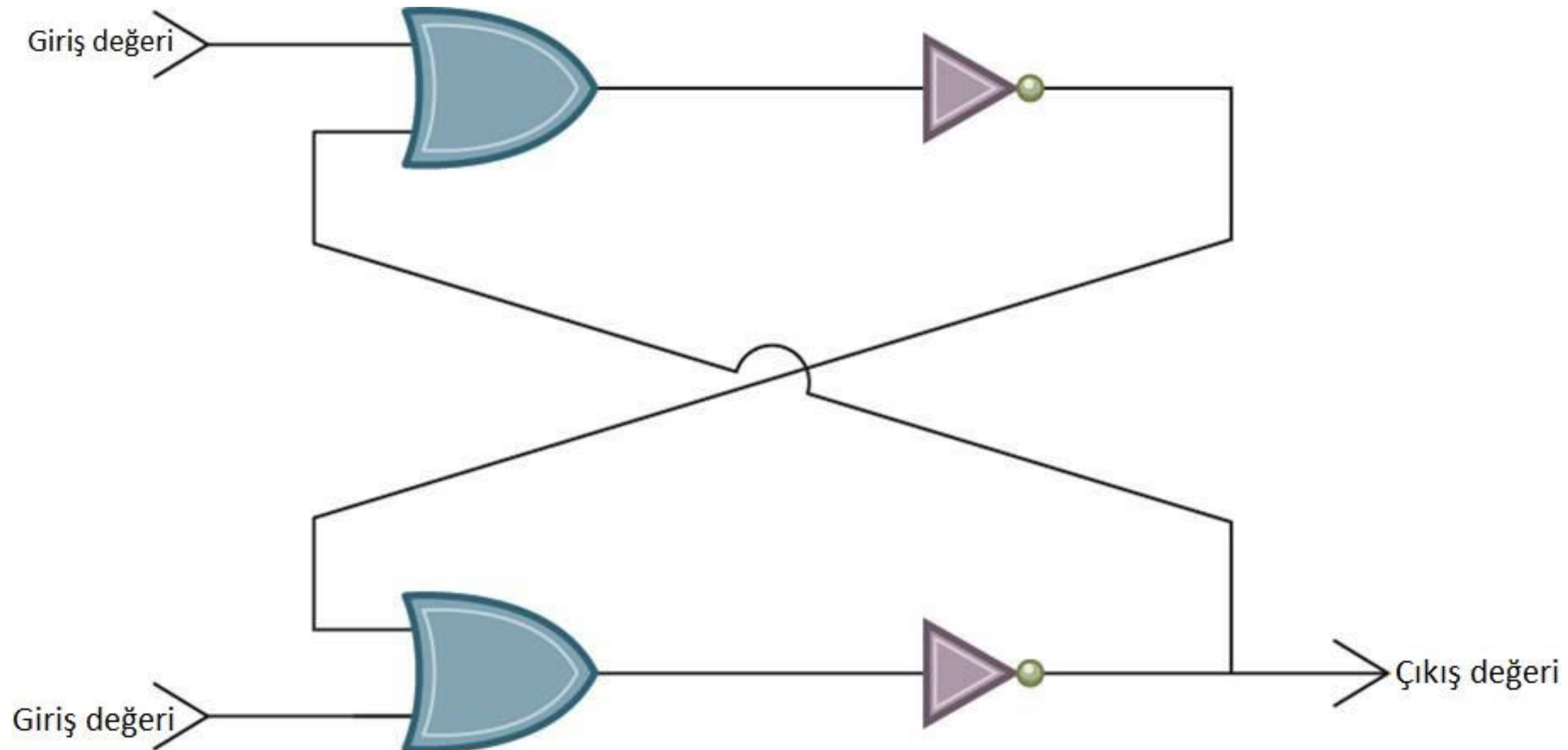


Şekil 1.4 Bir flip-flop'un çıkış değerini 1'e ayarlama (devamı)

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



Şekil 1.5 Bir flip-flop oluşturmanın 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

8 bit 'lik 7 yazılımı (bininiz bilgisayarın bit)

z² z¹ z⁰

0 0 0 0 0 1 1 1

Arabic Numerals
9 1 2 3 4 5 6 7 8 9
↳ Terci hint

2'lik ten → 16'lik sisteme
1986
7C2

Şekil 1.6 Onaltılık(Hexadecimal) Sayı Kodlama Biçimi

* Sayı dönüşümleri BİL!

Desimal
(DEC)

Hexadesimal (HEX)

0	10
1	11
2	12
3	13
4	14
5	15
6	16
7	17
8	18
9	19
A	1A
B	1B
C	1C
D	1D
E	1E
F	1F = 31

BINARY

0001 0111
1 7

HEX

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

OCT (8'lik)

09 10 10

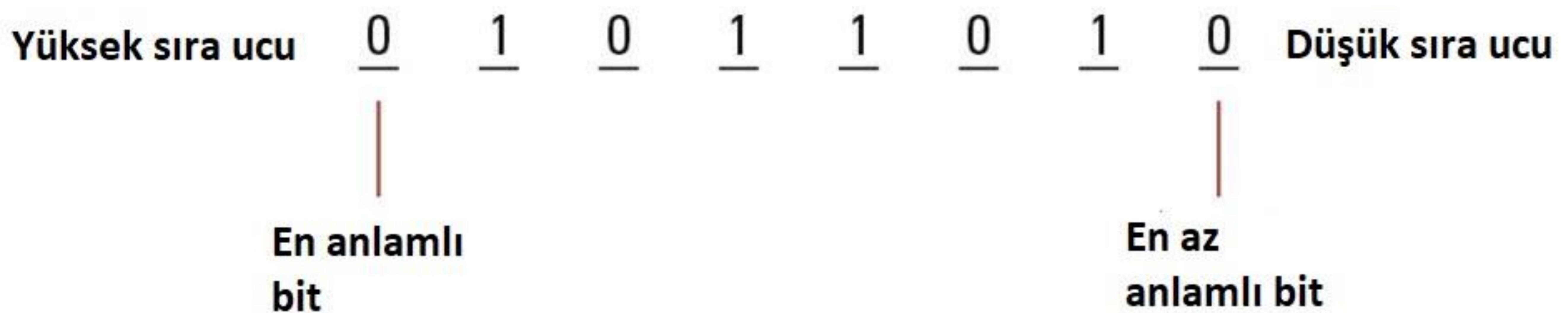


1.2 Ana Bellek

- **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)

En alt depolama birimi (byte)
1 harf 1 byte

Şekil 1.7 Bir byte(bayt) boyutundaki bellek hücresinin düzeni



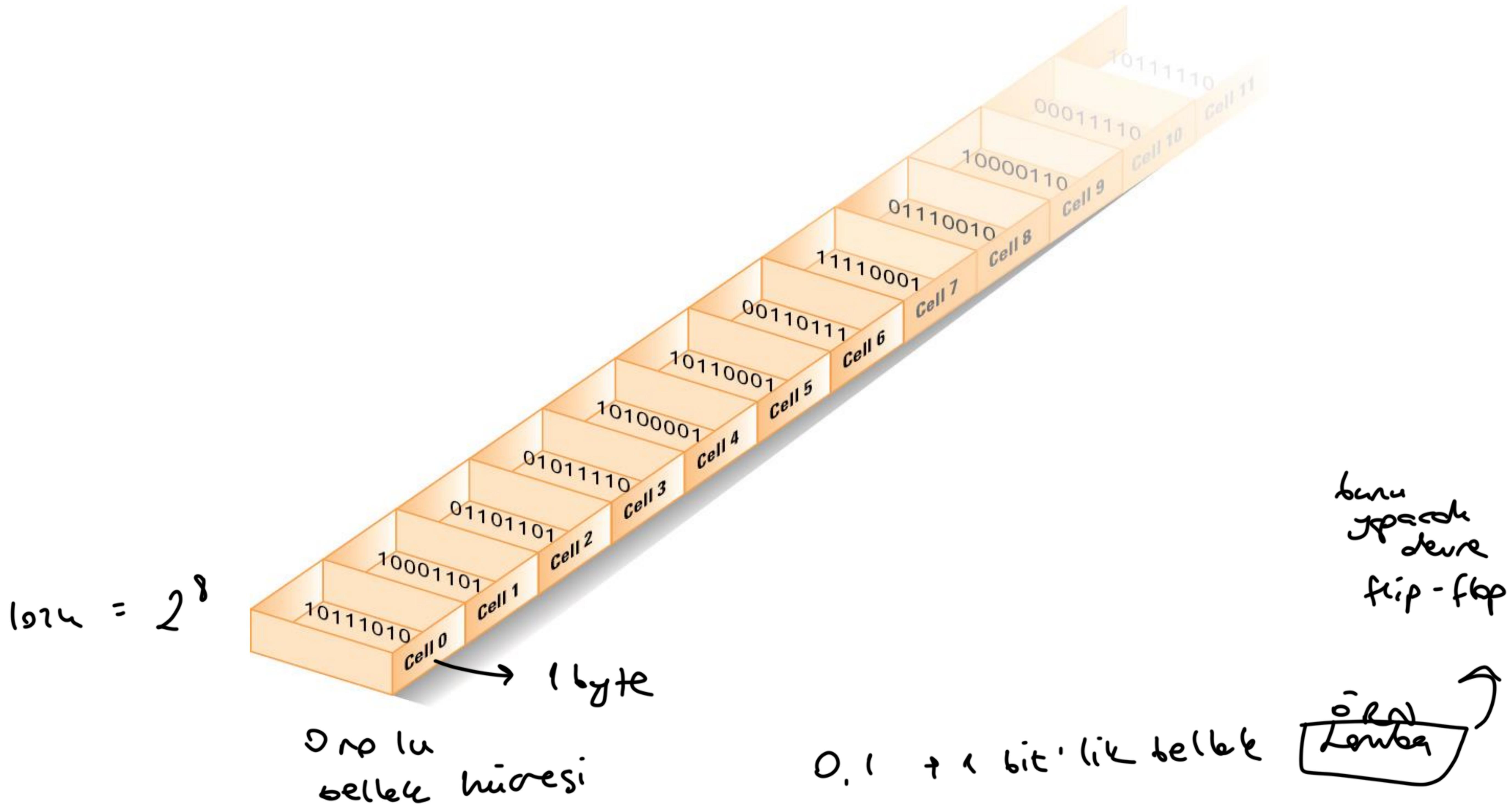
Ana Bellek Adresleri

- **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.

256 FF

1 2 3 4 5 6 7 8 9 A B C D E F

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

Elektrik
vrsa bilgi var

Hard Disk'te rastgele erişimli değildir.

SSD içinde kullanılarak RAM gibi, sabit yedek disk'

SSD disk

metin

kimya
yapı

Resetlenen denek belleğinin boşaltılmış durum

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

Disk üreticisi 1000'in katları
olarak hesaplar satar.
Bilgisayar 1024 üzerinde

1.3 Yığın Depolama

- Ek aygıtlar:
 - Manyetik diskler
 - CDler
 - DVDler
 - Manyetik kasetler
 - Flaş sürücü
 - Katı hal diskleri *SSD*
- 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 isteğiinin gönderilmesi ve verinin gelmesi arasında geçen zaman

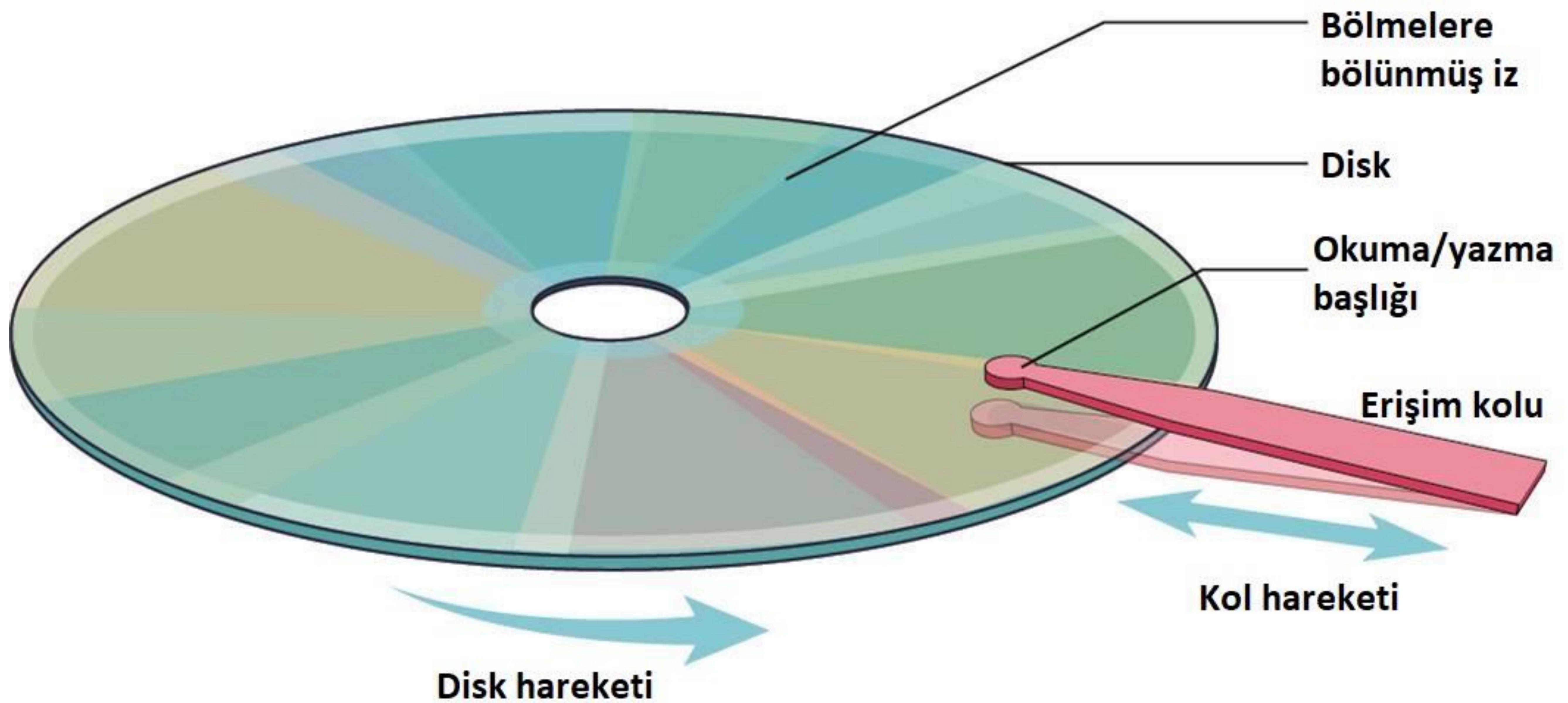
SSD'de
üzerindeki tipin
kalitesi

SSD diskten veri kurtarması
zordur.

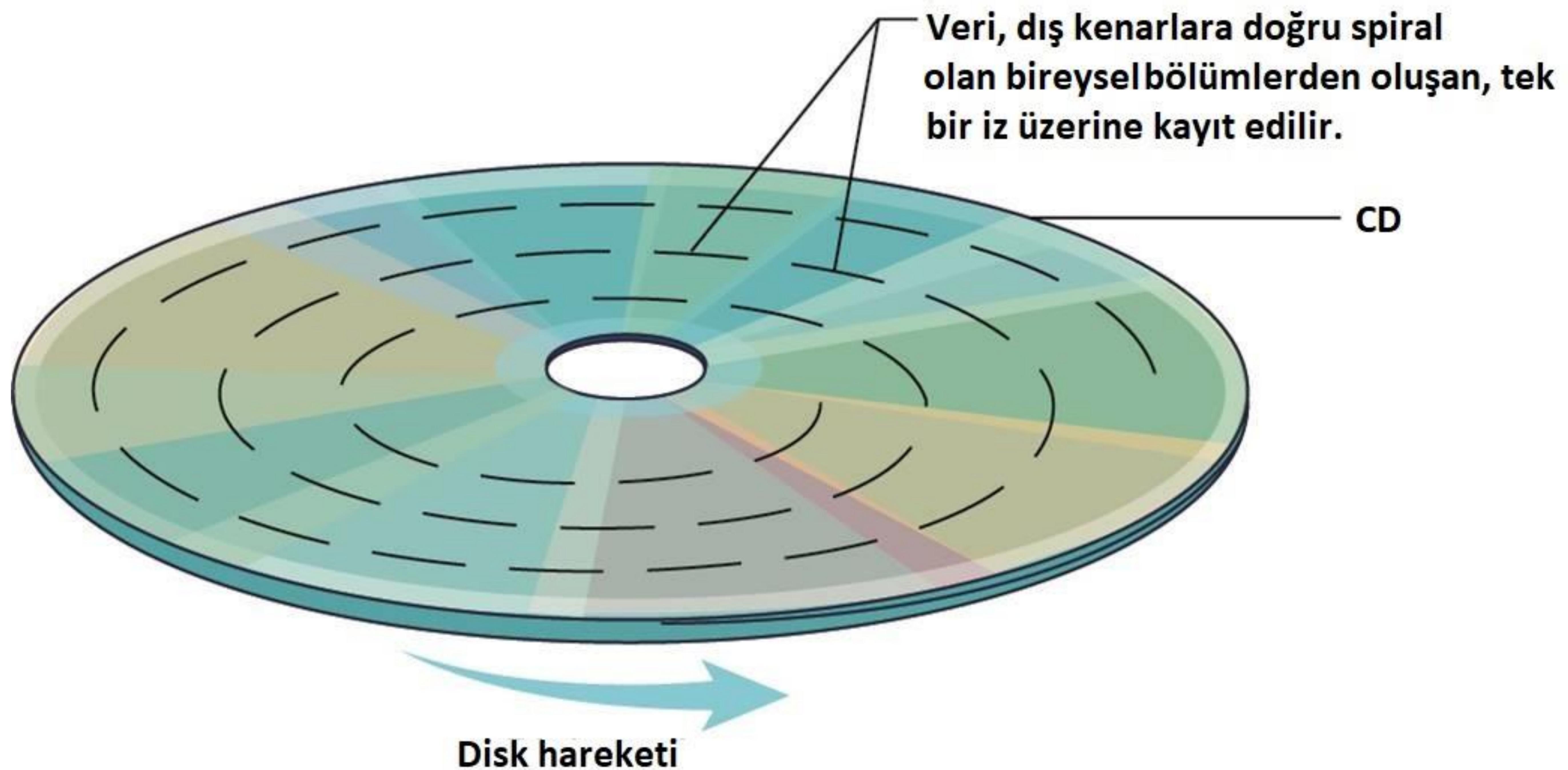
Klasik diskten bilgi kurtarması
daha kolay

İşletim sistemi görevi diskte
nereye kaydedeceğini seçmek,
gizlemek.

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. (*SSD de de bu durum vardır.*)
- **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 işaretleri, vb.) eşsiz bir bit desenine sahiptir.**
 - **ASCII:** İngiliz alfabetesindeki simbol ve ifadelerin gösterimi için 7-bit deseni kullanır.
 - **Unicode (UTF-8):** Dünya genelinde kullanılan dillerdeki simbol ve ifadelerin gösterimi için 16-bit 21-bit arası desenleri kullanır

ascii Table

Şekil 1.11

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

01001000	01100101	01101100	01101100	01101111	00101110
H	e	I	I	o	.

C'de
dili yapmak

Java'da Python'da
string yapısı

Sayısal Değerlerin Gösterimi

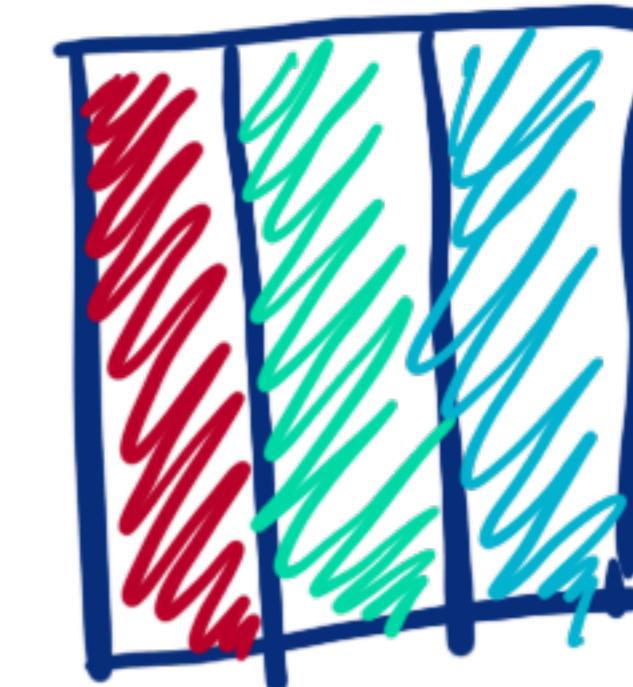
- **İkilik Sayı Sistemi:**
Sayıları bitler şeklinde göstermek için 2 rakam kullanır
 - Bilgisayardaki bütün sayısal değerler 1ler ve 0lar dizisi şeklinde saklanır, depolanır.

C dilinde 1b'lik olurğunu ifade etmek için başına 0x kütlesini

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



1 piksel

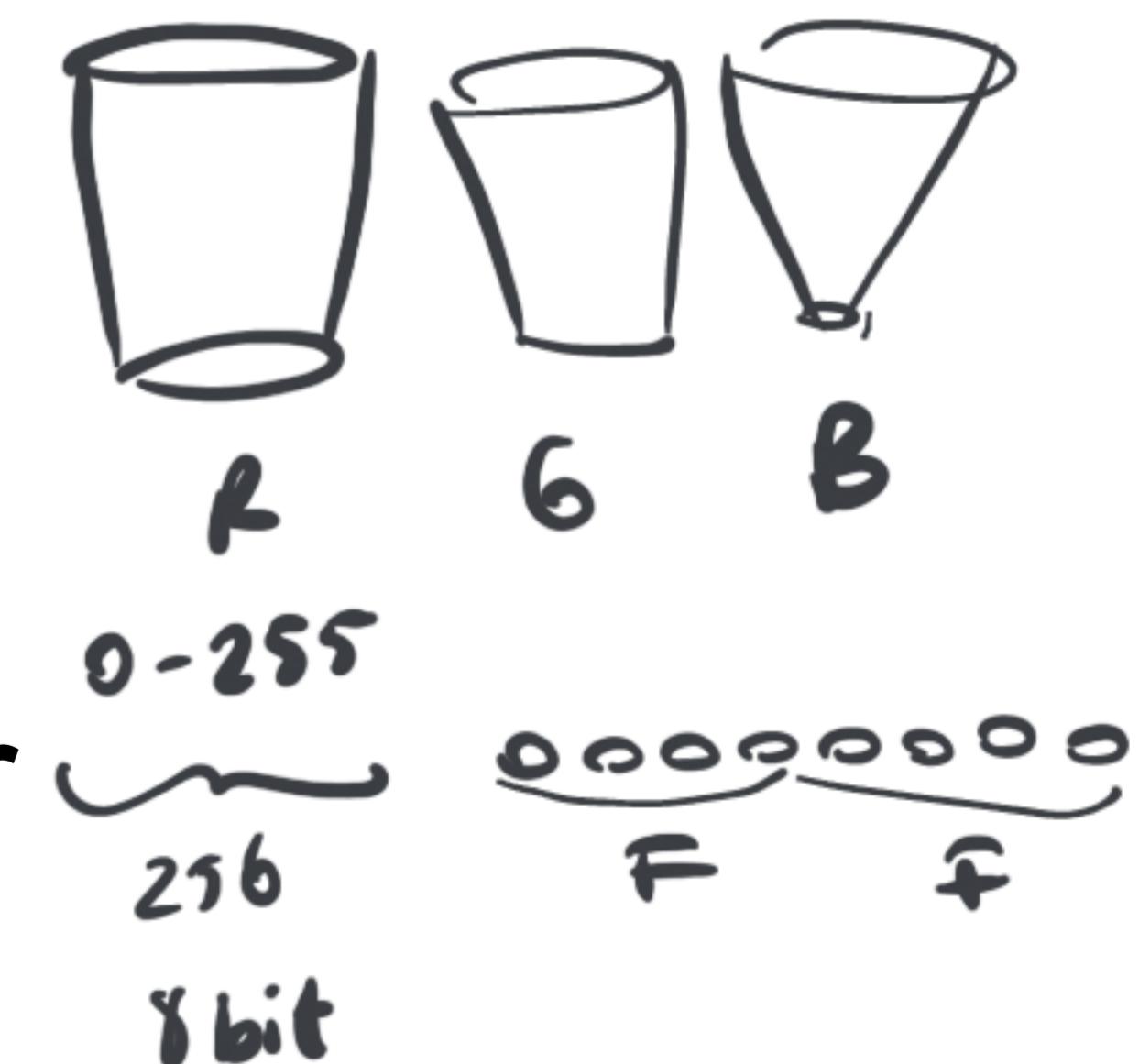
Verilen voltaj
noktaya göre at
got yarar.

matrisin içinde
küçük düşügeklide
bululur.

* Ekran kartı
işlevcisi
+ matris hesabı

burun skin var.

RGB Formatı



Resmi vektör dörek tutma

Görkein, egrinin doğrusun
derikleri dörek tutular (Görüntü işlemesi)

O zaman resmin kalitesi (zoom'da)
bozulmaz.

Renk paleti ferla işe gerek
daha ferla yer kaplayebilir.

Bileş.
Sayılıl
görül.

$$\begin{bmatrix} 01010000 \\ 01110000 \\ 10010000 \\ 10101111 \end{bmatrix}$$

Sesin Gösterimi

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

Ses aslında bir dalgası

Ses telefon veya bilī. Kayıt ederken

Bellīli zaman aralıklarında sesin
siddetini ölçer

Ses, ısı, sıçratıcı analog (surelli)
Sinyallerdir.

Yazıcıda kullanılmaz
gündüz kağıt siyah
değil

CMYK }
y a e } derin
g g i } renk
n h l } formate
t w }

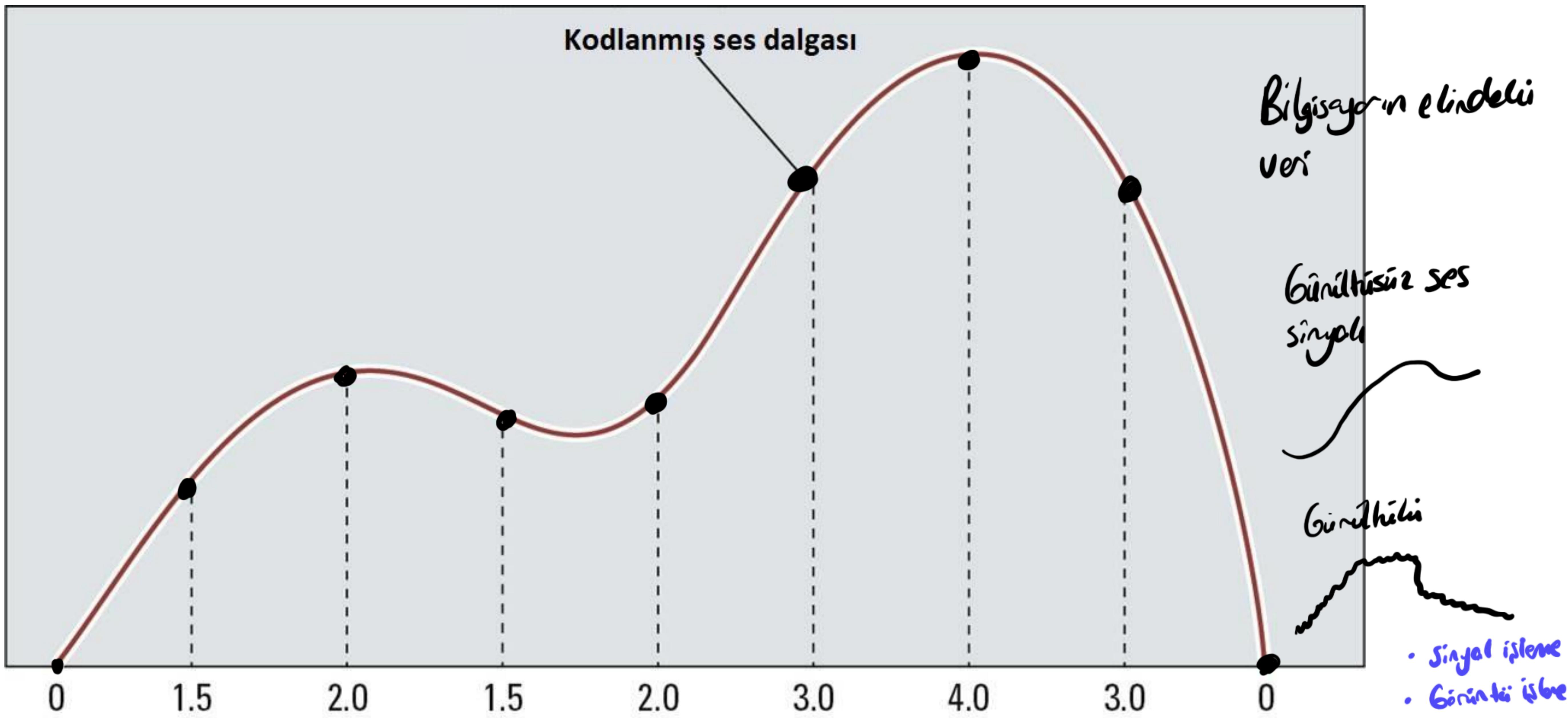
Bilgisayarda Zoom

4 7 3
2 5 1
6 4 9



44 77 33
44 77 33 medyalesme
22 55 11
22 55 11 sonuncu
sayfa
66 44 99
66 44 99

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



Plaklar analog kaydeden.

Bilg. ayrık sinyalleri saklayabilir (Ayrık Kapılar Dersi)

Genlikler

Filmde ses ve görüntü ayrı ayrı kaydedilir.
ayrı ayrı dosyada bulunur. ama aynı aynı kaydedilir.

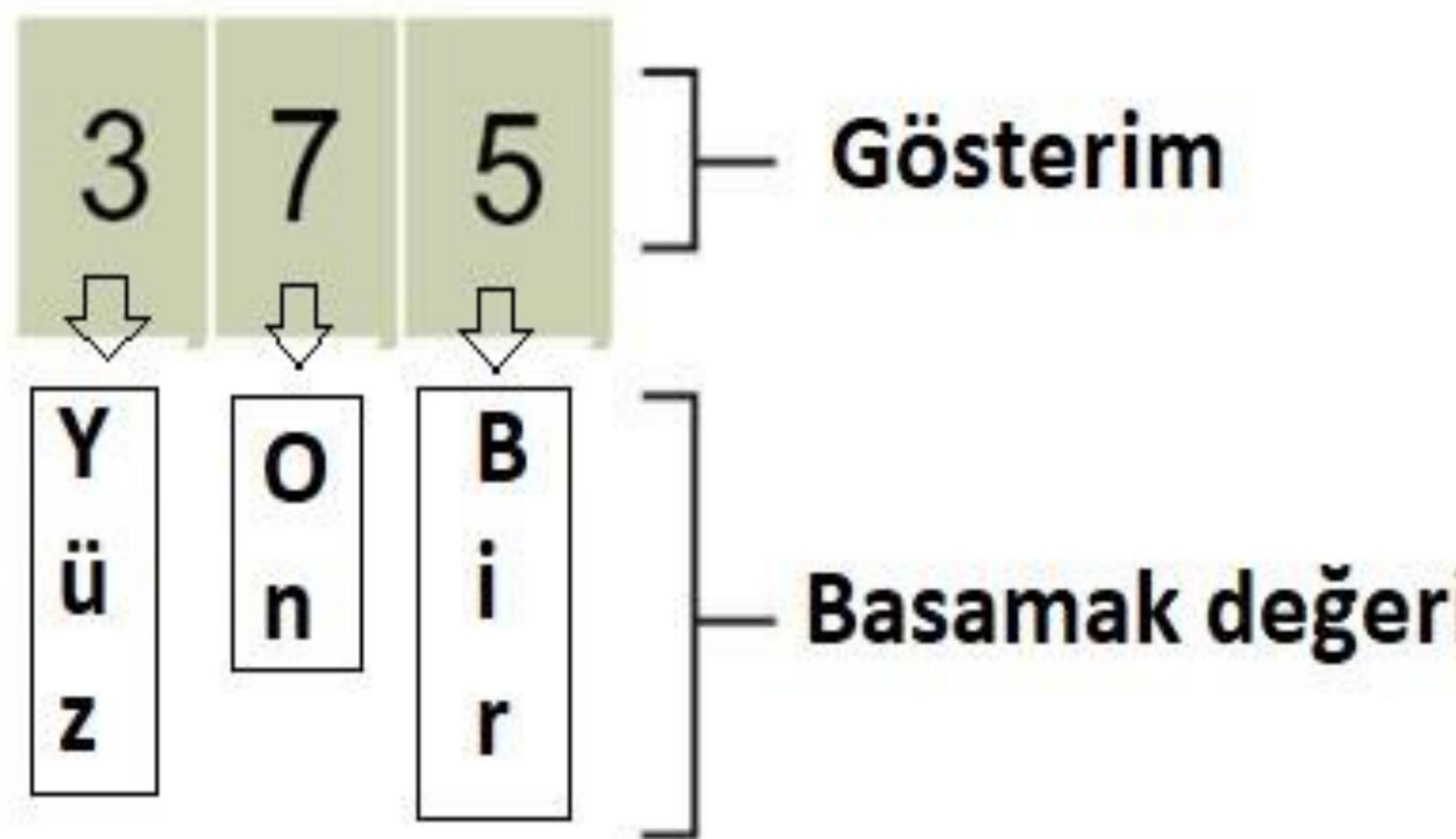
1.5 İkilik Sistem

Geleneksel ondalık sistem, onun kuvvetleri üzerine kurulmuştur.

İkilik Sistem ise ikinin kuvvetleri baz alınarak oluşmuştur.

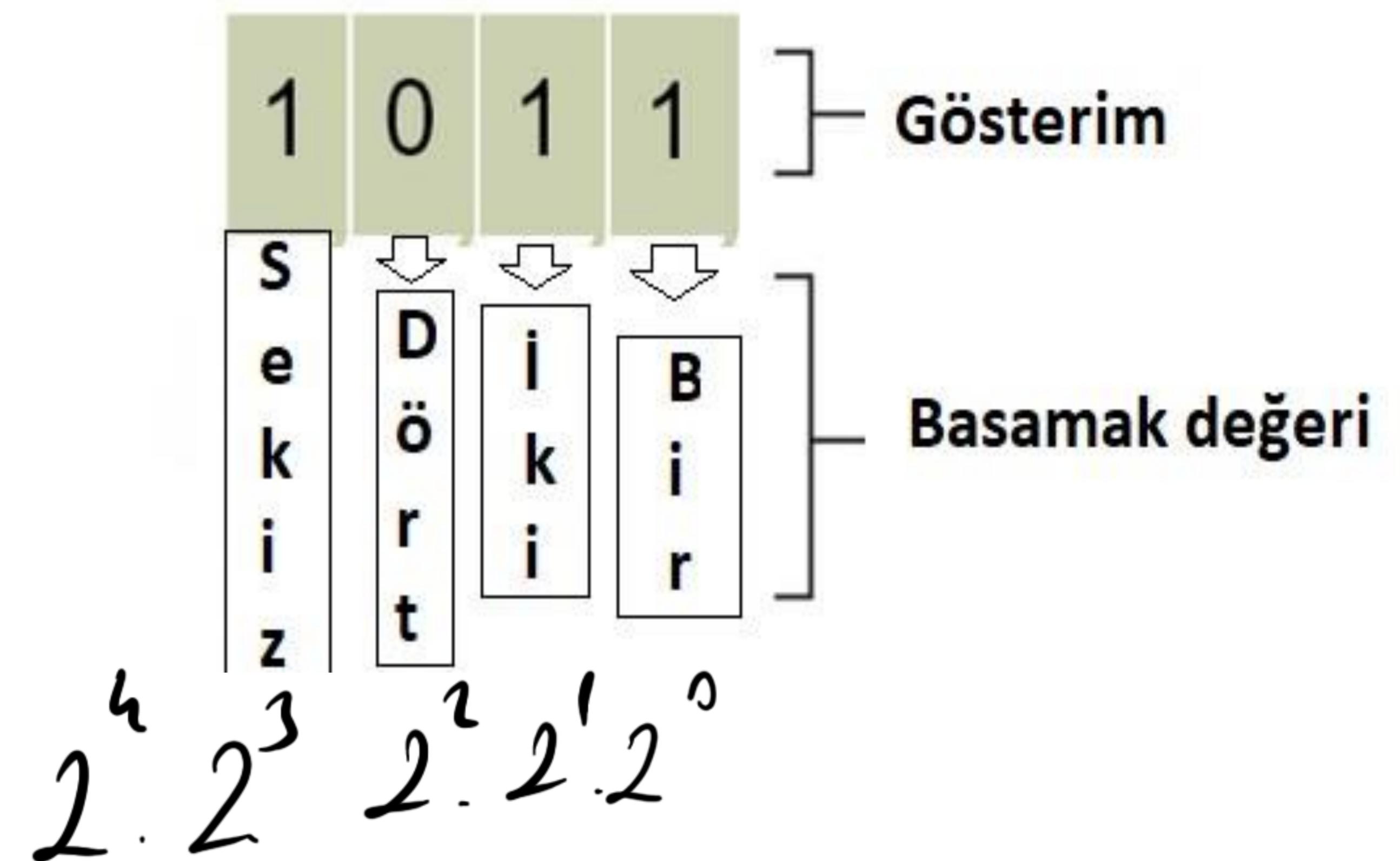
Şekil 1.13 Onluk ve ikilik taban sistemleri

a.Onluk taban sistemi

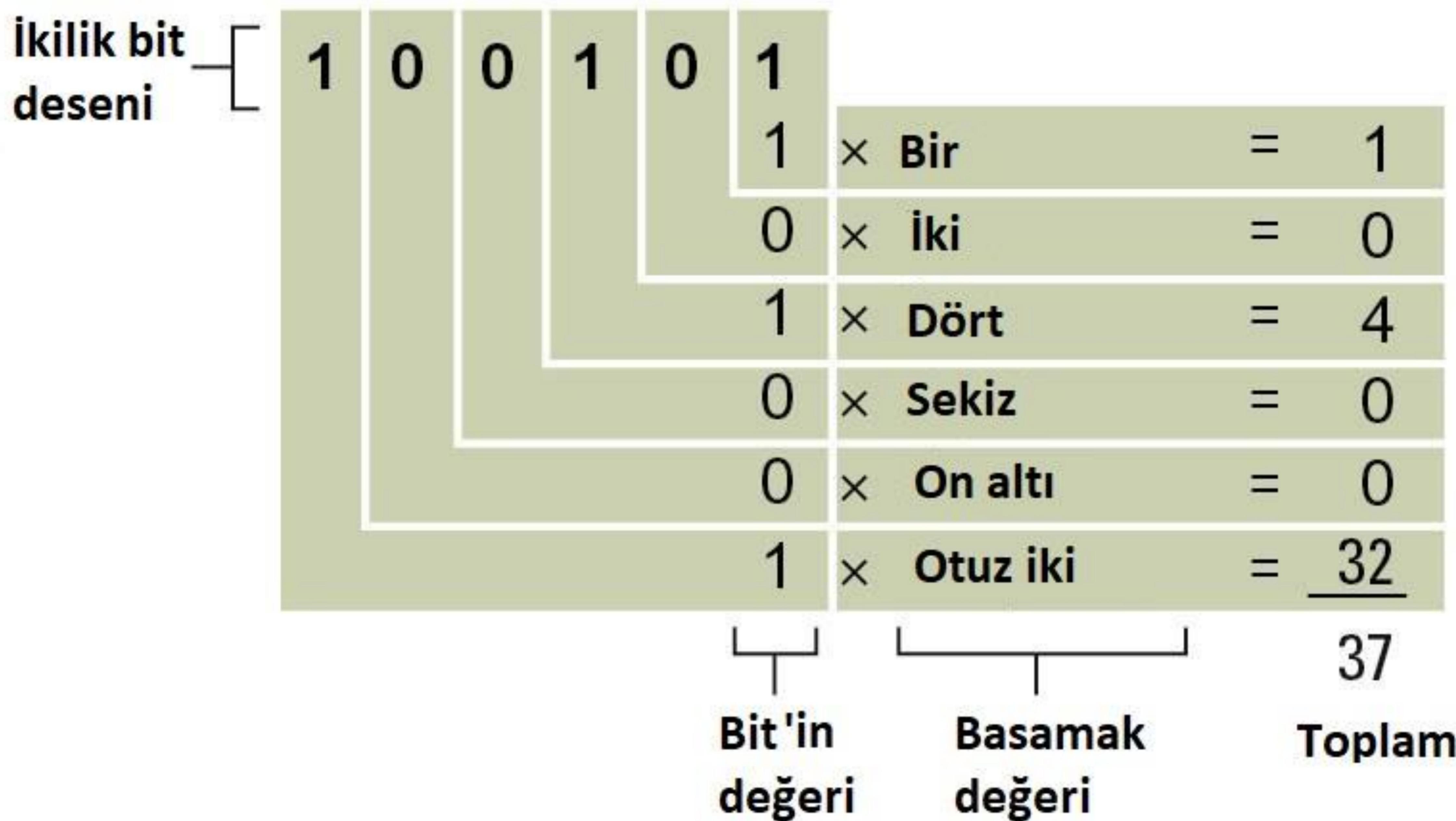


Basamak değeri ile matematik yapılıyorsa

b.İkilik taban sistemi



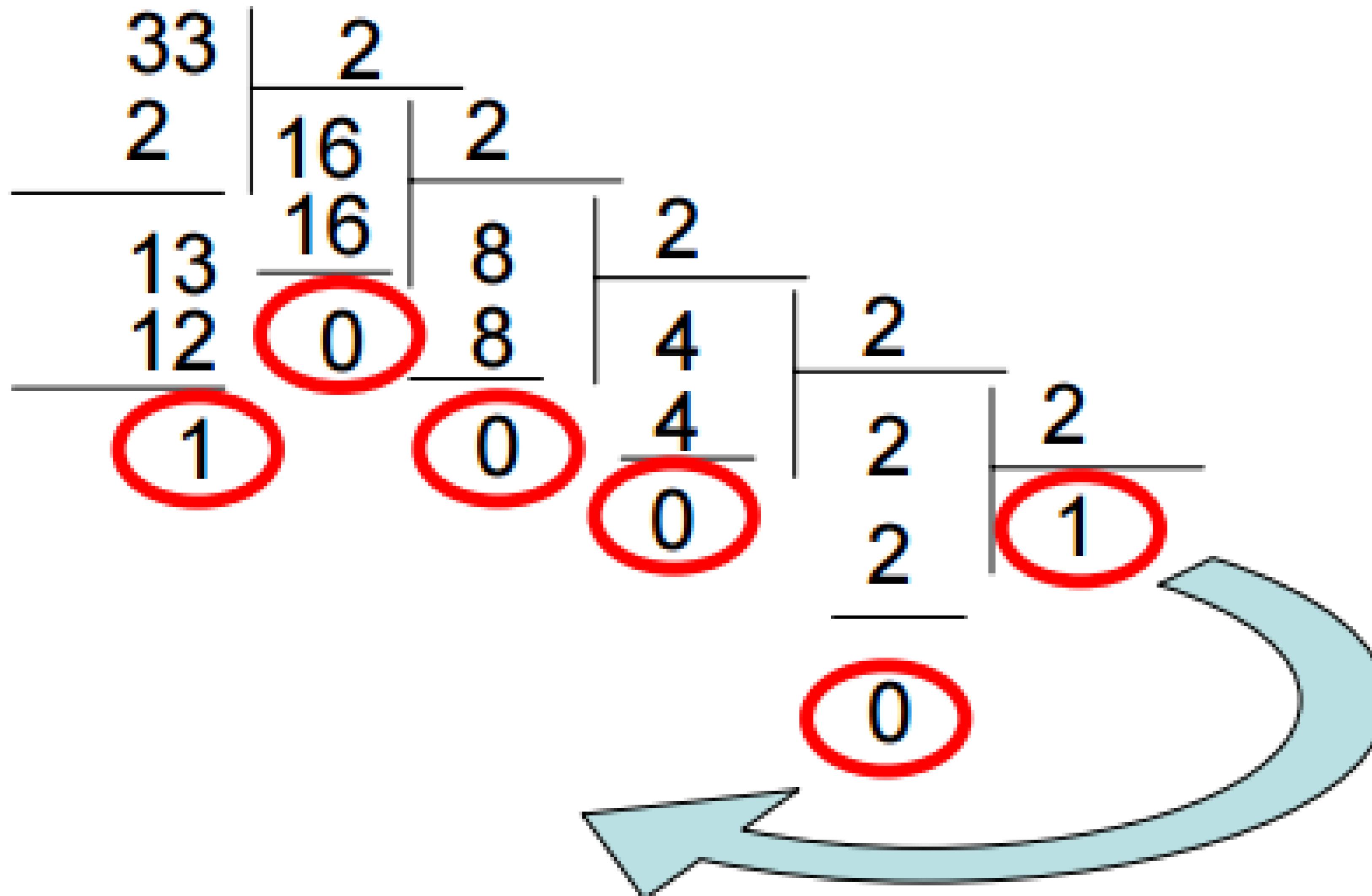
Şekil 1.14 100101 ikilik gösterimini çözümleme



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 değerin 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



$$(33)_{10} = (100001)_2$$

↙ Sırası Sonusu

$$(33)_{10} = (x)_2 \quad x=?$$

↓

2ye en yakın

$$32 = 2^5 \underset{1}{\cancel{1}} + 2^4 \underset{0}{\cancel{0}} + 2^3 \underset{0}{\cancel{0}} + 2^2 \underset{0}{\cancel{0}} + 2^1 \underset{0}{\cancel{0}} + 2^0 \underset{1}{\cancel{1}}$$

1 0 0 0 0 1

3 bitlik

$$\begin{array}{r} 101 \rightarrow 5 \\ + 011 \rightarrow 3 \\ \hline 1000 \rightarrow 8 \end{array}$$

Şekil 1.17 İkilik sistemde toplama

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

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

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

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

8-6.2

$$\begin{array}{r} 00000101 \\ + 00000011 \\ \hline 00001100 \end{array}$$

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

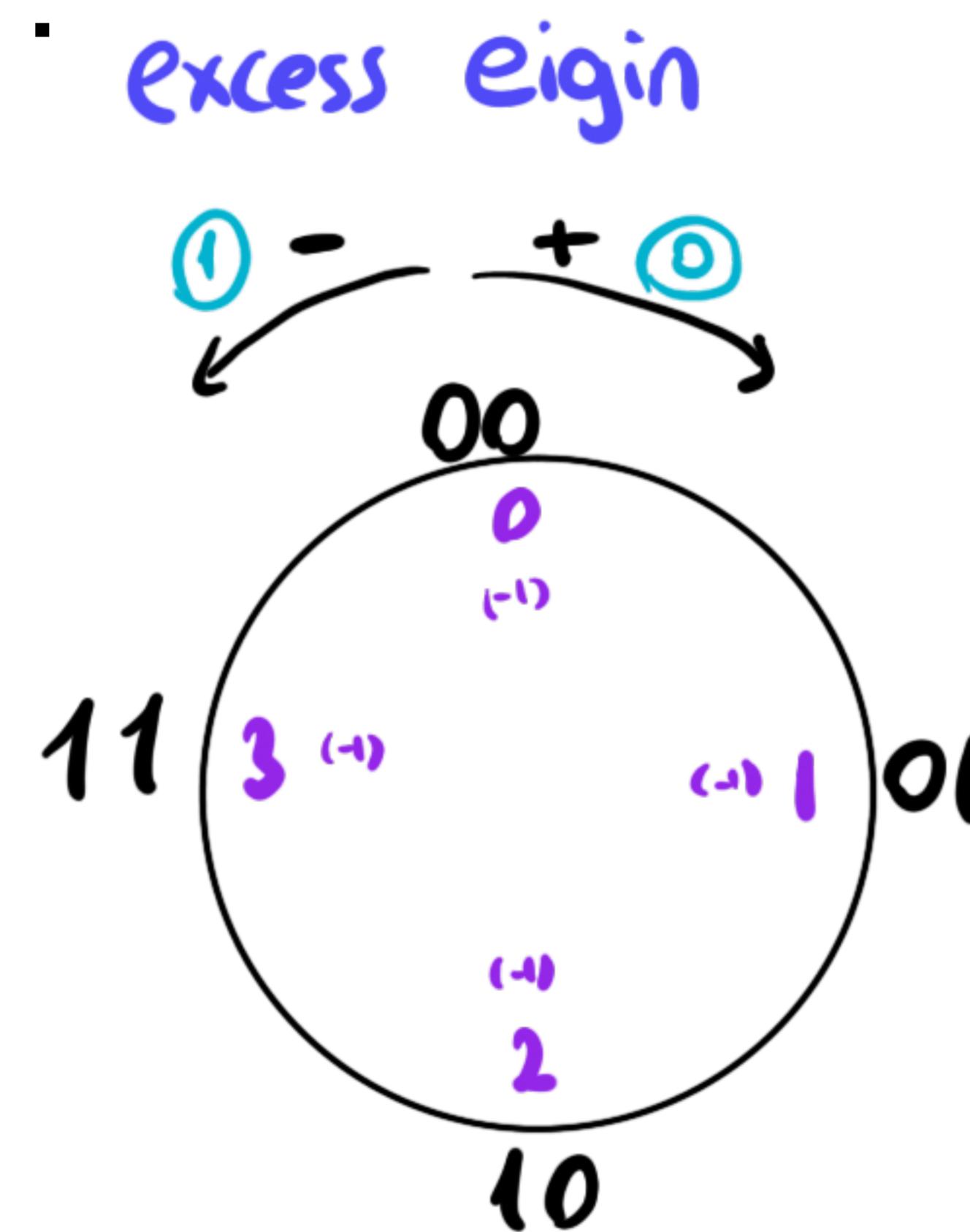
Hayatta en çok serdiğim 1. kişi jestin
0 bulgısın.

Şekil 1.18 101.101 ikilik gösteriminin çözümlenmesi

1.6 Tam sayıların depolanması

- İkinin tümleyeni gösterimi: Tam sayıların en popüler gösterimidir. *2's complement*
- Fazlalık (Excess) gösterimi: Tam sayıların diğer bir gösterim şeklidir.

Bir sayının negatifini bulmak için



ince bükün $1 \rightarrow 0$
 $0 \rightarrow 1$

$00000011 \rightarrow 3$
 $0000001000 \rightarrow$
101 10110
 \downarrow
 -3

Şekil 1.19 İkinin tümleyeni gösterimi

O negatif oladığı için
-1er + larla 1 fazladır

short
-32.368 +32.767

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

negatif sayının
ilk biti her
zaman 1
pozitif sayının
ilk biti her zaman 0

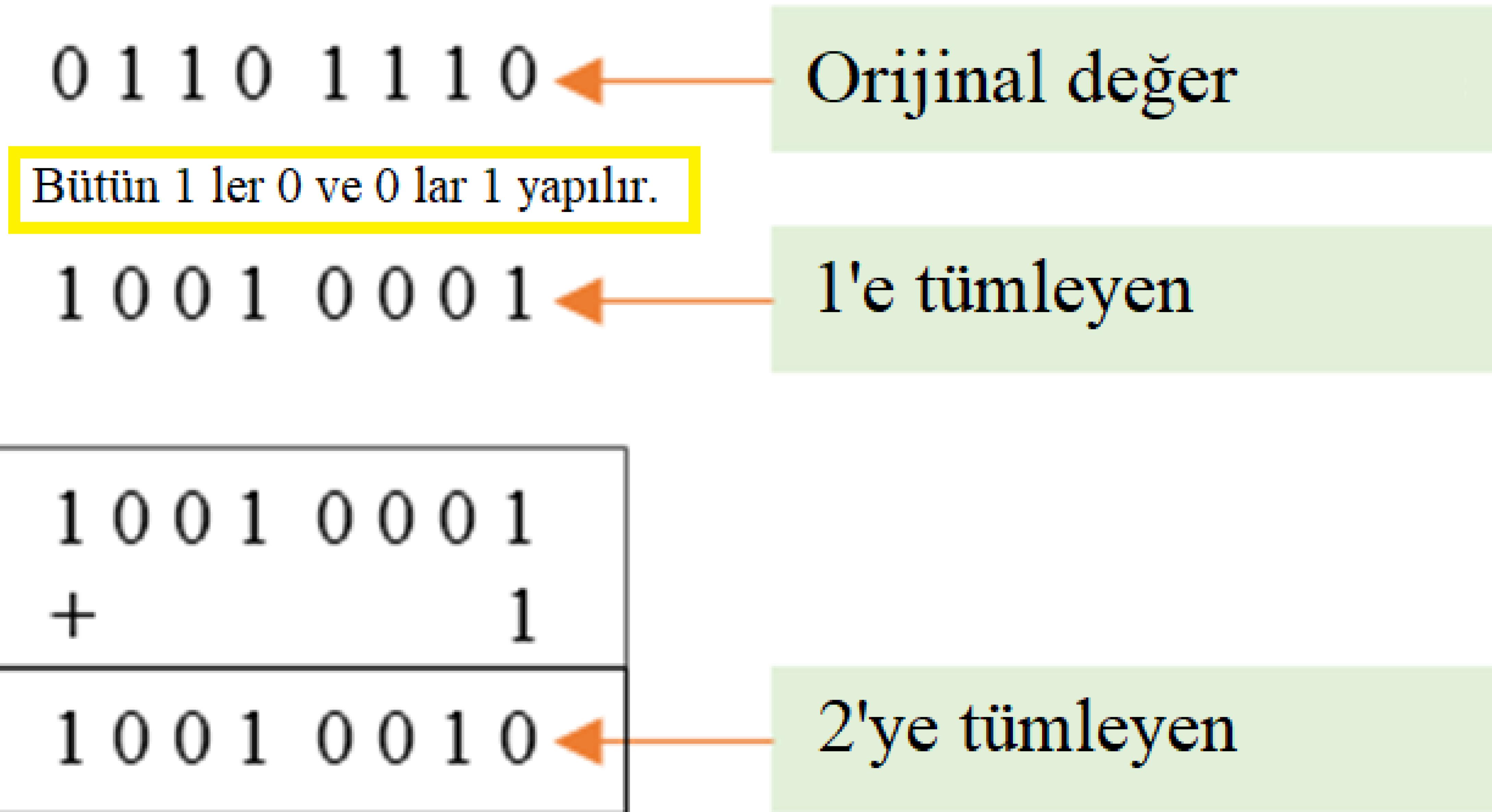
Signed (İşaretli)
ise

unsigned → işaretli tutma
değişkenin değer aralıkları
kontrol et & sınırlar
gökmeli

Java, C++ sınır kontrolü
yapar hata verir.

+ Bilg. on kütüphane 8 bit

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



Şekil 1.21 İkinin tümleyeniyle ilgili alıştırmalar

$0010 + 2$
 $1010 - 2$
 desem

On tabanında işlem

İkinin tümleyeninde işlem

10 tabanında sonuç

Bilg. mikro işlemci yaparken
 * toplama yapar devreyi
 çıkarma için de kullanabilir
 miyim?

3+2

3-2

3+(-2)

Daredeki ekr.
 tozumut

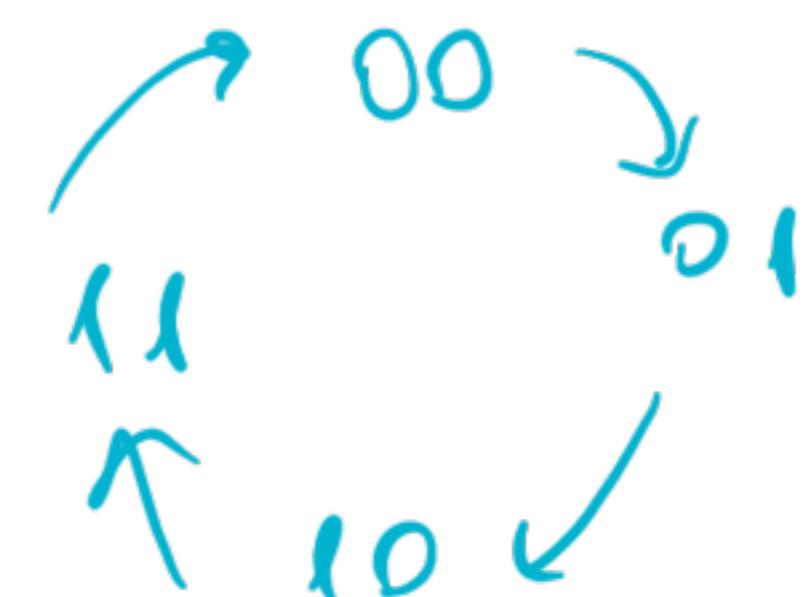
2's complement

$\begin{array}{r} 3 \\ + 2 \\ \hline \end{array}$	\rightarrow	$\begin{array}{r} 0011 \\ + 0010 \\ \hline 0101 \end{array}$	\rightarrow	5
$\begin{array}{r} -3 \\ + -2 \\ \hline \end{array}$	\rightarrow	$\begin{array}{r} 1101 \\ + 1110 \\ \hline 1011 \end{array}$	\rightarrow	-5
$\begin{array}{r} 7 \\ + -5 \\ \hline \end{array}$	\rightarrow	$\begin{array}{r} 0111 \\ + 1011 \\ \hline 0010 \end{array}$	\rightarrow	2

Taşma Problemi (Overflow)

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

maximuma değere ulaştığında



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

pozitifler

negatifler

GNU gcc açık
kodlu
kaynak
tarayıcı

MSUC

Microsoft C++
Macbook Linux, C,
Intel C compiler

Compiler
ASSEMBLY ile
yazılır
İşlevciye göre değişir

Java'ın code onası
hangi işlevci olursa
olsun oradaki
katmanı
kaldırır

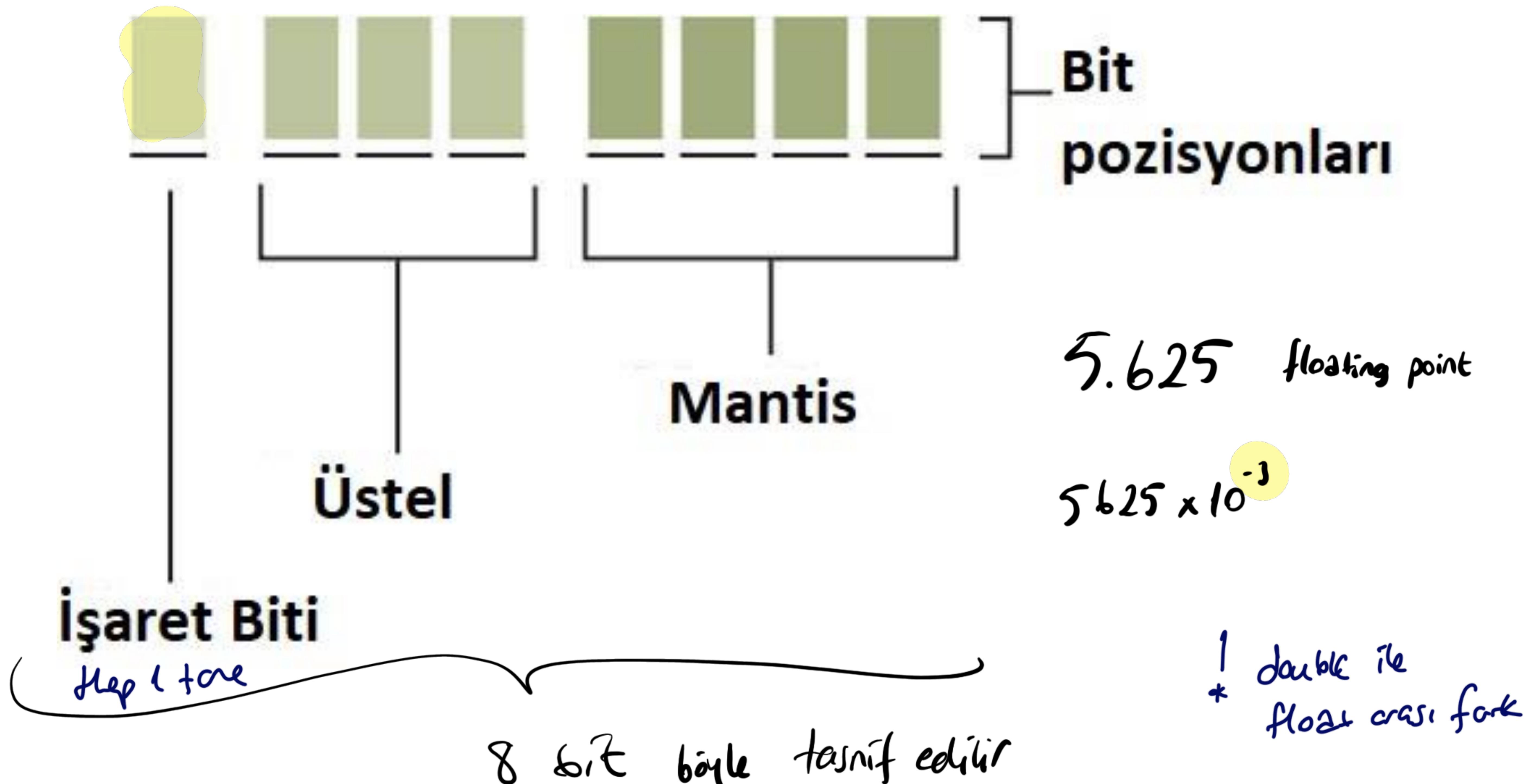
Şekil 1.23 Üç bit uzunluğundaki desenleri kullanan «fazlalık sekiz» gösterim sistemi

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

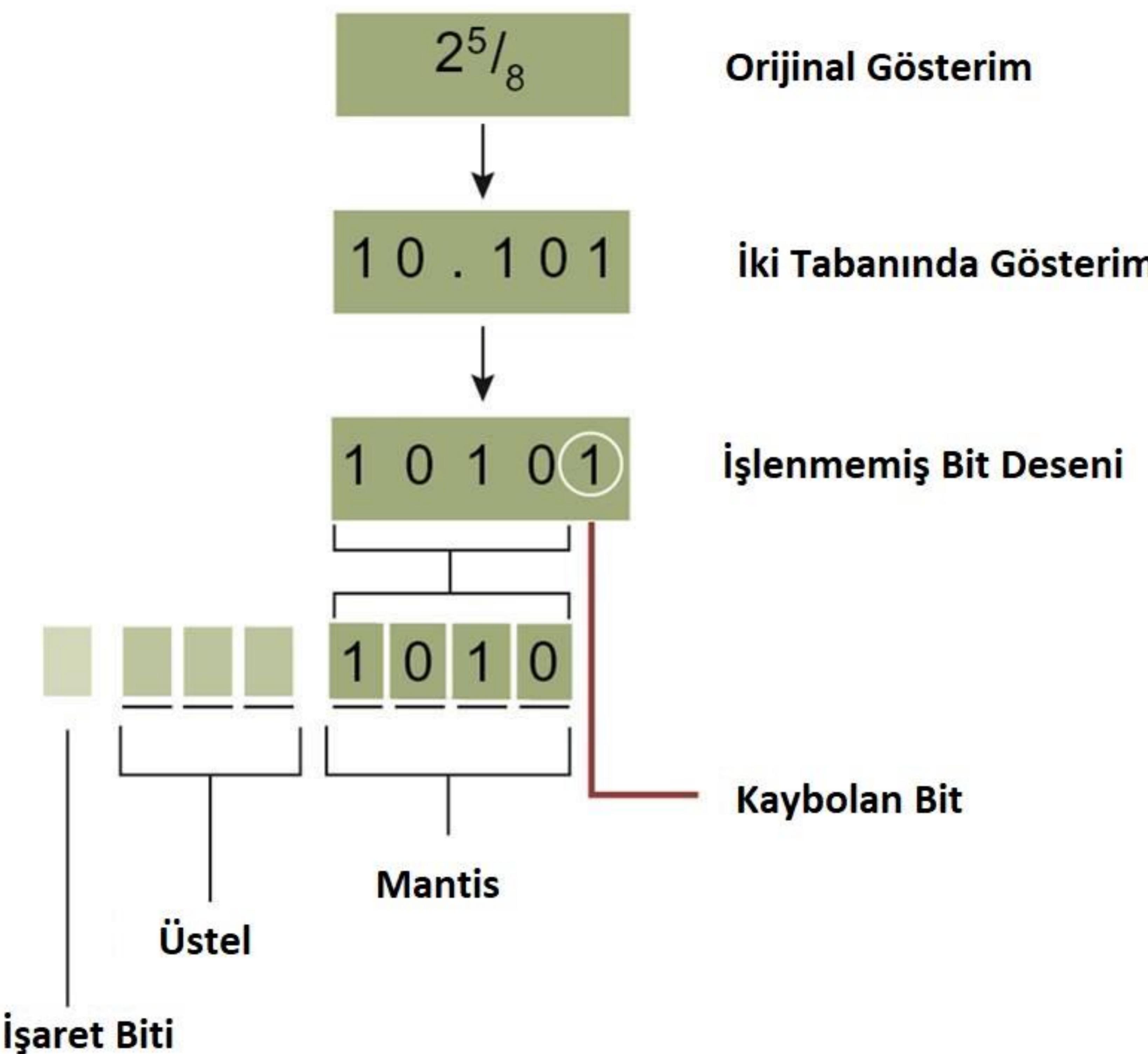
1.7 Kesirleri Depolama

- **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



Devirli sayılar da

Code

0.9 ~ 1 e

Yerde

Kodluşuge ceviri

Kırpma (Yuvarlama) Hataları 0.3

- 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

Uygulama C

```
#include <stdio.h>
int main() {
    float x=1;
    float y=3;
    float sonuc = x+y;
    printf("%f",sonuc);
```

float
beldede söyle tutulur 1.00000
int x=1 1

1.8 Veri ve Programlama

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

Python

python.exe
programleur

Notepad-i aç

a=5
b=3
print(a+b)

script tabanlı

derlene diye bir şey yoktur. (C'de void)

verdiğin değer enveldir.

↳ derler derlediği exe 'yi
galerdir.

C de yaptiğın
re comunitàn
ASSEMBLY Kodlarını
göndürsin

Python ile işletim
sistemi yararlı

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 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
```

OB binary , 00 oktan

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
                  üs alma
```

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

Para Birimi Dönüşümü

```
# Para birimi dönüştürme için bir program.

USD_to_GBP = 0.66      # Bugünün döviz kuru
GBP_sign = '\u00A3'    # £ simgesi 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))

# Sayıyı metin haline getirince karakterlerle işlem yaplanır
```

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

Bu kod
kural olarak
değin, çalışıyor

matik hatası



1.9 Veri Sıkıştırma

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

coding encoding

ne kadar az şey gönderirsen
o kadar hızlı gidiyor

10MB resim 2MB
gelişmez

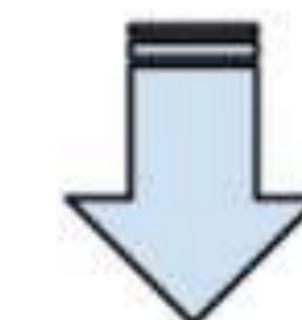
2MB word
word zip
kayıpsız

Run-length encoding

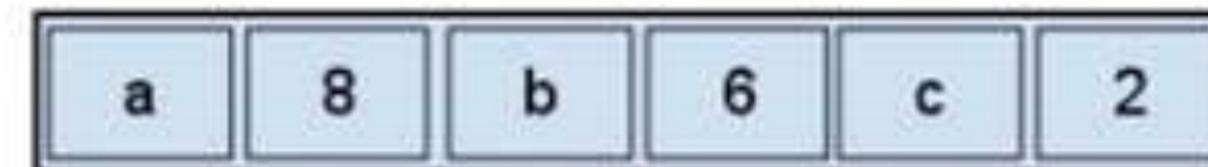
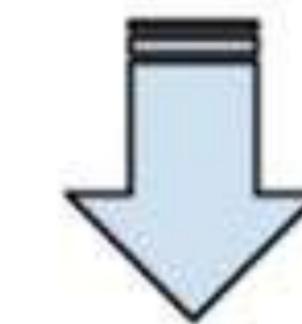


Run-length encoding is most useful when data repeats itself often.

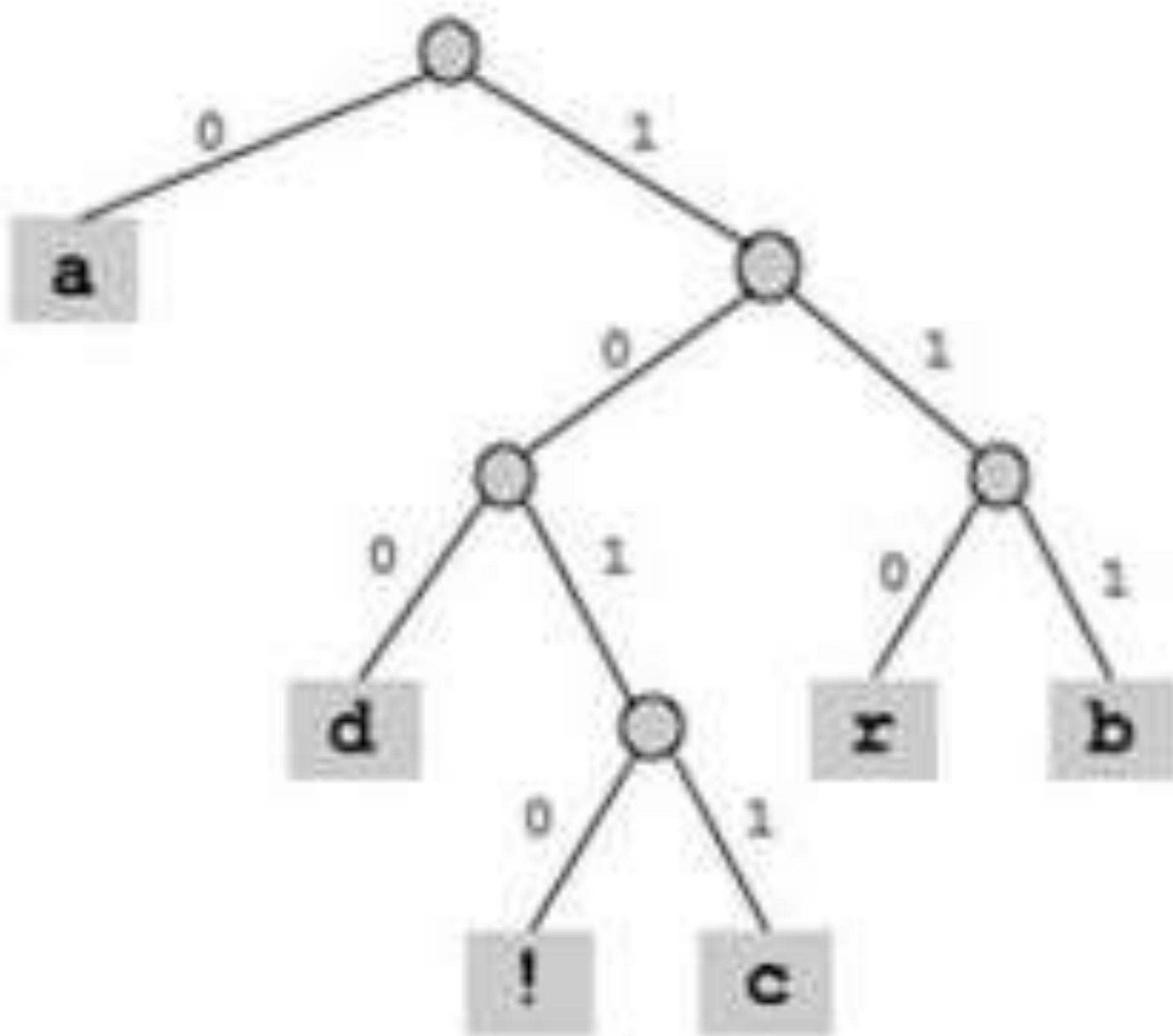
This immediately means that it is not well suited to text – as there aren't many words that contain more than two of the same letters repeated next to each other.



run-length encoding



HUFFMAN CODE DATA COMPRESSION



char	encoding
a	0
b	111
c	1011
d	100
x	110
!	1010

Relative Encoding

- Relative Encoding:

1 2 3 4	1 3 3 4	0 1 0 0
2 5 3 7	2 5 3 7	0 0 0 0
3 6 4 8	3 6 4 7	0 0 0 -1
4 7 5 9	3 7 5 9	-1 0 0 0
1 st Frame	2 nd Frame	Difference

Resulting **difference** can be RLE.

depİzeleri gönderir tasarıf eder

Dictionary-based encoding example

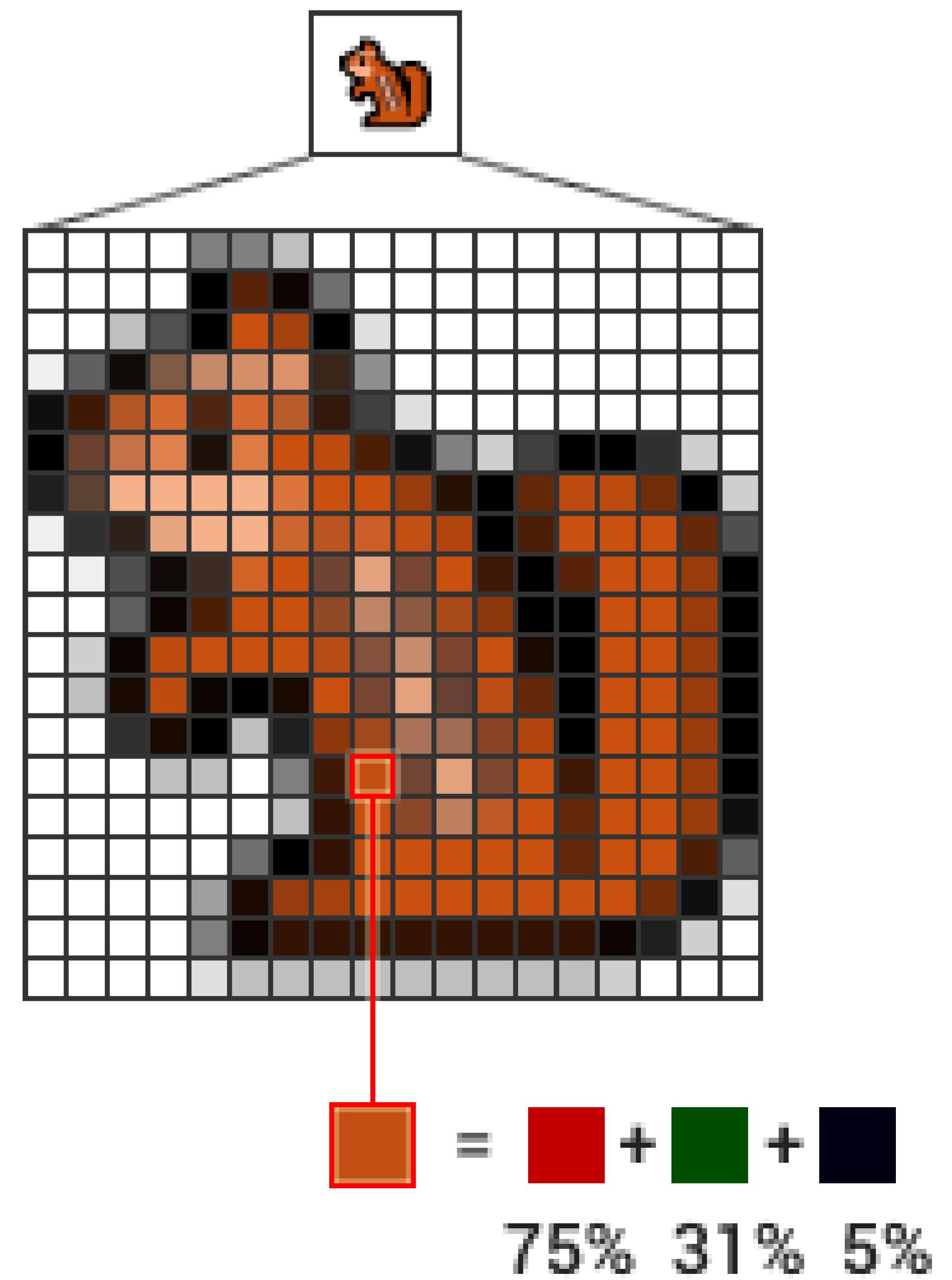
- Dictionary:
 1. ASK
 2. NOT
 3. WHAT
 4. YOUR
 5. COUNTRY
 6. CAN
 7. DO
 8. FOR
 9. YOU
- Original text:
 1. ASK NOT WHAT YOUR COUNTRY CAN DO FOR YOU ASK WHAT YOU CAN DO FOR YOUR COUNTRY
- Encoded based on dictionary :
 - 1 2 3 4 5 6 7 8 9 1 3 9 6 7 8
4 5

ASK her haf kodlusa 2n bit
ASK → 1 diye kodla 1 bit

Görüntüleri Sıkıştırma

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

bmp
formatı noda
kayıtlıdır



Ses ve Video Sıkıştırma

- MPEG
 - Yüksek kalite televizyon yayını
 - Video konferans
 - MP3
 - Ses
 - Müzik
- ogg
acık kaynak
kodlu sıkıştırma
algoritması

1.10 İletişim Hataları

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

o b e r e l e l o l l y d ö n n e r s m a t h e o ' s
d e n e .

Şekil 1.26 Tek eşlik için A ve F kodlarının ayarlanması

Gönderileceği zaman 1 bit eklenir.



Nesinin bozuk olduğunu bilenin ama bozuk olduğunu bilmek.
-lerin gönder.

Veri Sıkıştırma Nedir?

Veri sıkıştırma, verilerin boyutunu küçültmek amacıyla uygulanan işlemlerdir. Bu, verilerin depolanmasını ve iletimini daha verimli hale getirir. İki temel veri sıkıştırma yöntemi vardır:

- 1. Kayıpsız Sıkıştırma:** Verinin orijinal haline tamamen geri döndürüleceği sıkıştırma yöntemidir. Örneğin, Huffman Kodu, Run-Length Encoding (RLE) ve Lempel-Ziv-Welch (LZW) gibi teknikler kullanılır [\[1\]](#).
- 2. Kayıplı Sıkıştırma:** Verinin bazı kısımlarının kaybedildiği, ancak genel kalitenin kabul edilebilir düzeyde kaldığı sıkıştırma yöntemidir. JPEG (görüntü) ve MP3 (ses) gibi formatlar bu yöntemi kullanır [\[1\]](#).

1.9 7. SORU

-sayısal bilgiyi görüntüleri, sesleri bit desenleri olarak Kodlama yapıılırken ortak olan sıkıntılı bir fenomeni tanımlayınız



✓ Sizin için yanıtlar oluşturuluyor...

Sayısal bilgiyi görüntüler, sesler ve bit desenleri olarak kodlarken karşılaşılan ortak bir sıkıntılı fenomen, **simgelerarası girişim** (Inter-Symbol Interference, ISI) olarak bilinir. Bu fenomen, bir sinyalin birden fazla bitinin birbirine karışması sonucu oluşur ve bu da veri iletiminde hatalara yol açabilir ¹.

ISI, özellikle yüksek hızda veri iletiminde ve dar bant genişliğine sahip kanallarda önemli bir sorun haline gelir. Bu durum, bitlerin birbirine karışmasına ve alıcı tarafında doğru bir şekilde çözülememesine neden olur. ISI'yi azaltmak için çeşitli teknikler kullanılır,örneğin:

- Eşitleyiciler:** Alıcı tarafında sinyalin bozulmasını düzeltmek için kullanılır.
- Kanal Kodlama:** Hataları tespit etmek ve düzeltmek için ek bilgi ekler.
- Filtreleme Teknikleri:** Sinyalin istenmeyen bileşenlerini azaltmak için kullanılır ¹.

Bu fenomenin etkilerini azaltmak, veri iletiminde güvenilirliği artırmak için kritik öneme sahiptir. Başka bir konuda daha fazla bilgi almak ister misiniz?

