

BURSA TEKNİK ÜNİVERSİTESİ  
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ  
BİLGİSAYAR MÜHENDİSLİĞİNE GİRİŞ DERSİ

**HAZIRLAYAN**

**İSİM-SOYİSİM :** Nazan Şal  
**NUMARA:** 25360859060  
**SINIF:** 1.sınıf

**DANIŞMAN**

**İSİM-SOYİSİM :** Turgay Tugan Bilgin

**KONU**

Bit ve Byte kavramları  
İkilik, Onluk ve Onaltılık Sistemler  
Negatif sayıların bilgisayarda tutulması

# İÇİNDEKİLER

- 1 : Kapak**
- 2: İçindekiler**
- 3 : Bit kavramı**
- 4 : Byte kavramı**
- 5-11 : Onluk sistem**
- 12-18 : İkilik sistem**
- 19-25 : Onaltılık sistem**
- 26-28 : Dönüşüm**
- 29-30 : Negatif tam sayı depolama**
- 31 : Kaynakça**
- 32 : Sonuç**

# ***BIT KAVRAMI***

**Bit (Binary Digit):** Dijital sistemlerde bilginin temsil edildiği en küçük veri birimidir. Bilgisayar içinde tüm veriler 0 ve 1 değerleriyle kodlanır ve bu değerler bit olarak adlandırılır.

**Tarihçesi** 1948 yılında Amerikalı matematikçi ve elektrik mühendisi Claude E. Shannon tarafından tanımlanmıştır.. Claude E. Shannon, iletişim sistemlerinde bilginin nicel olarak ölçülmesi, iletilmesi ve kodlanması ihtiyacını incelemiştir. Bu amaçla, bilgiyi en sade biçimde ifade edebilmek için iki durumlu (0 ve 1) bir yapı önermiş ve bu yapının temel birimini bit (binary digit) olarak adlandırmıştır. Bit kavramı, modern dijital iletişim ve bilgisayar bilimlerinin temelini oluşturmuştur.

# **BYTE KAVRAMI**

**Byte:** 8 bitten oluşan ve dijital sistemlerde veri depolamak için kullanılan temel veri birimidir.

**Tarihçesi:** 1956 yılında IBM'de çalışan bilgisayar bilimci Werner Buchholz tarafından tanımlanmıştır. Werner Buchholz, bilgisayar sistemlerinde verilerin adreslenebilir, anlamlı ve standart birimler hâlinde işlenebilmesi ihtiyacını incelemiştir. Bu amaçla, birden fazla bitten oluşan ve genellikle bir karakteri temsil edebilen bir veri birimi tanımlamış ve bu birime byte adını vermiştir. Byte kavramı, veri depolama, bellek adresleme ve karakter kodlama sistemlerinin temelini oluşturmuştur.

# *DECİMAL SİSTEM (onluk)*

**Tarihçesi:** Antik matematikçiler tarafından milattan önce( tam olarak zamanı bilinmiyor) ortaya atılmıştır. Aslen insanın on parmağına dayanır ve günlük hayatı kolaylaştırmak amacıyla kurulmuştur

## İşleyiş

4 3 2 1 0 -1 -2 -3

**12345.6789**

+ ← 0 → -

## Örnek

$$1746 = 1 \cdot 10^3 + 7 \cdot 10^2 + 4 \cdot 10^1 + 6 \cdot 10^0$$

3 2 1 0

# **Bilgisayarlarda Onluk (Decimal) Sistem Nerelerde Kullanılır?**

**Kullanıcı Arayüzleri:** İnsanların kolay anaması için sayılar onluk sistemde gösterilir.

**Giriş–Çıkış İşlemleri:** Klavyeden girilen sayılar ve ekranda görünen değerler onluk sistemdir.

**Programlama Dilleri (İnsan Tarafı):** Değişken değerleri ve hesaplamalar genellikle onluk olarak yazılır.

**Raporlama ve Sonuçlar:** Hesap makinesi sonuçları, istatistikler ve tablolar onluk sistemle sunulur.

**Hata Mesajları ve Ayarlar:** Port numaraları, sürüm numaraları ve ayar değerleri onluk sistemde ifade edilir.

# *Kullanıcı Arayüzleri*

**Onluk sistem, bilgisayarın kullanıcıya sunduğu arayüzlerde kullanılır.**

Bilgisayarlar kendi içinde ikilik sistemle çalışsa da, insanlar sayıları onluk sistemle daha kolay anlar.

Bu nedenle ekranada gördüğümüz tüm sayılar (dosya boyutu, saat, pil yüzdesi vb.) onluk sistemde gösterilir.

**Örnekler:**

Saat: 14:30

Pil: %85

Dosya boyutu: 120 MB

**Amaç:** Kullanıcının sayıları hızlı ve hatasız anlaması.

# *Giriş / Çıkış İşlemleri*

**Kullanıcıdan alınan ve kullanıcıya gösterilen sayılar onluk sistemdedir.**

**Klavyeden girilen sayılar onluk sistem olarak algılanır.**

**Bilgisayar bu sayıları önce ikilik sisteme çevirir, işlem yapar, sonra sonucu tekrar onluk sistemde gösterir.**

**Örnek:**

**Kullanıcı:  $25 + 15$  yazar**

**Bilgisayar: İşlemi binary olarak yapar**

**Ekran: Sonucu 40 olarak gösterir**

**Amaç:** Onluk sistem, kullanıcı ile bilgisayar arasında köprü görevi görür.

# *Programlama Dilleri (İnsan Tarafı)*

Programlama dillerinde sayılar onluk sistemle yazılır.

C, Python, Java gibi dillerde değişkenlere verdığımız değerler genellikle onluk sistemdedir.

Bu, programcının kodu daha rahat yazmasını ve okumasını sağlar.

**Örnek:** (C dili):

int yas = 20;

int not = 75;

Bilgisayar bu değerleri arka planda binary olarak işler.

**Amaç:** Onluk sistem, programcı için kolaylık ve okunabilirlik sağlar.

# *Raporlama ve Sonuçlar*

*Hesaplama sonuçları kullanıcıya onluk sistemle sunulur.*

*Bilgisayar içinde yapılan karmaşık işlemler sonunda çıkan sonuçlar, raporlar ve tablolar onluk sistemle gösterilir.*

**Örnek:**

*Sınav notları*

*İstatistik sonuçları*

*Hesap makinesi çıktıları*

*Finansal raporlar*

**Amaç:** Çünkü kullanıcı sonuçları onluk sistemde yorumlayabilir.

# *Hata Mesajları ve Ayarlar*

Bazı sistem bilgileri ve ayarlar onluk sistemle ifade edilir.

Bilgisayar ayarlarında ve hata mesajlarında sayılar çoğu zaman onluk sistemde gösterilir.

**Örnekler:**

Port numaraları (8080, 3000)

Sürüm numaraları (v2.1, v10)

Hata kodları (404, 500)

**Amaç:** Kullanıcının ve geliştiricinin sayıları kolayca tanıayıp hatayı anlamasıdır.

# BİNARY SİSTEM (ikilik)

Leibniz tarafından 1703'de bulundu. Hesaplamayı basitleştirmek amacıyla yapıldı. Sistem, 1 ve 0'a dayanır ve elektrik devrelerine uyumludur. Bu sistem bilgisayarın temelini oluşturur.

## İşleyiş

$$\begin{array}{r} 1 \ 1 \ 1 \ 1 \\ 8 \ 4 \ 2 \ 1 \\ \swarrow \ \swarrow \ \swarrow \\ x_2 \ x_2 \ x_2 \end{array}$$

## Örnek

$$\begin{array}{r} 1 \ 1 \ 0 \ 1 \\ 8 \ 4 \ 2 \ 1 \end{array} \rightarrow 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$$

# **Bilgisayarlarda İkilik Sistem Nerelerde Kullanılır?**

**Donanım Seviyesi:** Bilgisayarlar elektrik sinyalleriyle çalışır.

**1 = açık, 0 = kapalı** durumunu temsil eder.

**Bellek (RAM & Depolama):** Tüm veriler (yazı, sayı, resim) 0 ve 1 olarak saklanır.

**İşlemci (CPU):** Tüm hesaplamalar ve karşılaştırmalar ikilik mantık ile yapılır.

**Yazılımlar:** Programlama dilleri, çalışırken \*\*makine dili (binary)\*\*ne çevrilir.

**Internet ve Ağ:** Veri传递 ikiilik kodlar halinde gerçekleşir.

# ***Donanım Seviyesi***

İkilik sistem, bilgisayar donanımının temel çalışma mantığıdır.

Bilgisayarlar elektrik sinyalleriyle çalışır. Donanım parçaları elektriğin var ya da yok olmasını algılar.

Bu iki durum ikilik sistemde:

1 = elektrik var

0 = elektrik yok

Transistörler yalnızca bu iki durumu tanıtabildiği için bilgisayarın en temel dili ikilik sistemdir.

# **Bellek (RAM ve Depolama)**

- Bilgisayardaki tüm veriler ikilik sistemle saklanır.
- Yazılıar, sayılar, resimler ve videolar bellek üzerinde 0 ve 1 dizileri olarak tutulur.
- Bellekteki en küçük birim bittir.

1 bit → 0 veya 1

8 bit → 1 byte

**Örnek:**

Bir harf veya sayı bellekte ikilik karşılığıyla yer alır.

# *İşlemci (CPU)*

- İşlemci tüm işlemleri ikilik sistemle gerçekleştirir.
- Toplama, çıkarma, karşılaştırma gibi işlemler CPU içinde mantık kapıları (AND, OR, NOT) ile yapılır.
- Bu kapılar yalnızca 0 ve 1 değerleriyle çalışır.

**Örnek:** Bir karşılaştırma sonucu:

1 → doğru

0 → yanlış

Bu sayede bilgisayar karar verebilir.

# *Yazılımlar ve Programlar*

- Tüm yazılımlar çalışırken ikilik sisteme çevrilir.
- Programcı kodu yüksek seviyeli bir dille yazar (C, Python vb.).
- Bu kod derleyici veya yorumlayıcı tarafından makine diline, yani ikilik sisteme dönüştürülür.
- Bilgisayar yalnızca 0 ve 1'lerden oluşan komutları anlayabilir.

# İnternet ve Veri İletimi

- İnternet üzerinden gönderilen tüm veriler ikilik sistemle iletilir.
- Mesajlar, e-postalar, fotoğraflar ve videolar paketlere ayrılır.
- Bu paketler ağ üzerinden 0 ve 1 şeklinde gönderilir.
- Alıcı taraf bu ikilik verileri tekrar anlamlı bilgiye dönüştürür.

# *HEXADECİMAL SİSTEM(ONALTILIK)*

İkilik sayıların daha kısa ve okunur biçimde ifade edilmesi amacıyla 20.yy'da bilgisayar mühendisleri tarafından geliştirildi. Sistemde 4 bitin 1 hex. basamağına karşılık gelir ve bilgisayar mimarisinde kullanılır.

## İşleyiş

0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F şeklinde 16 karakter bulunmaktadır

## Örnek

$$\begin{array}{r} 6 \ F \ E \ 3 \\ 3 \ 2 \ 1 \ 0 \end{array} \rightarrow 6 \cdot 16^3 + 15 \cdot 16^2 + 14 \cdot 16^1 + 3 \cdot 16^0$$

# **Bilgisayarlarda 16'lık (Hexadecimal) Sistem Nerelerde Kullanılır?**

**Bellek Adresleri:** RAM ve bellek adresleri daha kısa ve okunabilir olduğu için 16'lık sistemle gösterilir.

**Makine Kodu ve Assembly:** Düşük seviyeli kodlar binary yerine hexadecimal olarak ifade edilir.

**Hata Ayıklama (Debugging):** Hata kodları ve bellek içerikleri hex formatında incelenir.

**Renk Kodları:** Web ve grafik uygulamalarında renkler hex kodlarla gösterilir.

# *İkilik Sistemin Kısaltılmış Gösterimi*

Onaltılık sistem, ikilik sayıların daha kısa ve okunabilir yazımı için kullanılır.  
İkilik sistemde sayılar çok uzundur.

Onaltılık sistem, her 4 bit'i tek bir karakterle temsil eder.

Binary: 11111111

Hex: FF

Bu sayede uzun binary dizileri daha kolay okunur ve yazılır.

# *Bellek Adresleri*

Bellek adresleri genellikle onaltılık sistemle gösterilir.

RAM'deki adresler binary olarak tutulur ancak bu adresleri okumak zor olduğu için işletim sistemleri ve geliştiriciler hexadecimal gösterimi kullanır.

Örnek:

0x7FFFA3C0

Amaç: Daha kısa, daha düzenli ve hata yapma ihtimali daha düşüktür.

# *Düşük Seviyeli Programlama (Assembly)*

Makine kodu ve Assembly dili onaltılık sistemle ifade edilir.

Makine komutları binary olarak çalışır.

Ancak geliştiriciler bu komutları hexadecimal olarak yazar ve okur.

**Amaç:** Hex, binary ile birebir uyumludur ama insan için çok daha pratiktir.

# *Hata Ayıklama (Debugging)*

Hata ayıklama sırasında veriler onaltılık sistemle incelenir.

Debugger ve hex editor gibi araçlarda:

- Bellek içerikleri
- Register değerleri
- Hata noktaları

genellikle hexadecimal olarak gösterilir.

Bu, yazılım hatalarının daha hızlı tespit edilmesini sağlar.

# ***Renk Kodları (Grafik ve Web)***

Renkler onaltılık sistemle ifade edilir.

Web tasarımda renkler RGB değerlerinin onaltılık karşılığıyla yazılır.

**Örnek:**

#FF0000 → Kırmızı

#00FF00 → Yeşil

#0000FF → Mavi

Her iki hex karakteri, bir renk bileşenini temsil eder.

# DÖNÜŞÜMLER

$$\begin{array}{r} \begin{smallmatrix} 1 & 6 & 8 & 4 & 2 & 1 \end{smallmatrix} \\ (01001) \\ \hphantom{(01001)}_2 \end{array} = \begin{array}{r} (9) \\ \hphantom{(9)}_{10} \end{array}$$

$$\begin{array}{r} \begin{smallmatrix} 1 & 6 & 8 & 4 & 2 & 1 \end{smallmatrix} \\ (10101) \\ \hphantom{(10101)}_2 \end{array} = \begin{array}{r} (21) \\ \hphantom{(21)}_{10} \end{array}$$

$$\begin{array}{r} \begin{smallmatrix} 1 & 6 & 8 & 4 & 2 & 1 \end{smallmatrix} \\ (11010) \\ \hphantom{(11010)}_2 \end{array} = \begin{array}{r} (26) \\ \hphantom{(26)}_{10} \end{array}$$

İkilik tabandan  
onluk tabana  
dönüş örnekleri

# DÖNÜŞÜMLER

$$\begin{array}{r} \text{8 4 2 1} \\ \text{(1010)}_2 = (\text{A})_{16} \end{array}$$

$$\begin{array}{r} \text{8 4 2 1} \\ \text{(1101)}_2 = (\text{D})_{16} \end{array}$$

İkilik tabandan  
onaltılık tabana  
dönüş örnekleri

$$(\text{10110110})_2 = (\text{B6})_{16}$$

$$\begin{array}{r} \text{8 4 2 1} \\ \text{1011} \end{array} \quad \begin{array}{r} \text{8 4 2 1} \\ \text{0110} \end{array}$$

<b>Binary</b>	<b>Decimal</b>	<b>Hex</b>
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

# İkilik Onluk Onaltılık Tablo

# NEGATİF TAM SAYILARIN BİLGİSAYARDA TUTULMASI

## İKİNİN TÜMLEYENİ GÖSTERİMİ

*Negatif değerleri gösteren desenler uygun uzunlukteki 1'lerin bir dizi ile başlayarak ve daha sonra 1'leri izleyen tek bir 0'a erişilen bir desen elde edilinceye kadar ikilik olarak geriye sayılmakta. Bu desenler -1,-2,-3 değerlerini göstermektedik*

# *İkinin tümleyeni yöntemi örneği*

Bit deseni	Gösterdiği değer
0011	3
0010	2
0001	1
0000	0
1111	-1
1110	-2

# KAYNAKÇA

- **BİLGİSAYAR BİLİMİNE GİRİŞ - Computer Science An Overview**
- **Sayı Sistemleri Ders Notları, Bilgisayar Mühendisliği Bölümü, Bursa Teknik Üniversitesi**
- **Khan Academy, “Number Systems”**

# **SONUÇ**

**Bu projede, bit ve byte kavramları, negatif tam sayıların tutulması ve sayı sistemleri konusu ele alınmış; onluk, ikilik, sekizlik ve onaltılık sayı sistemleri arasındaki dönüşümler incelenmiştir. Yapılan örnekler ve açıklamalar sayesinde sayı sistemlerinin bilgisayar sistemlerinde neden ve nasıl kullanıldığı anlaşılmıştır. Özellikle ikilik ve onaltılık sayı sistemlerinin, bilgisayarların çalışma mantığı açısından önemi vurgulanmıştır. Proje kapsamında elde edilen bilgiler, sayısal verilerin bilgisayar ortamında temsil edilmesini ve dönüştürülmesini anlamada temel bir altyapı sağlamıştır.**