

## MİKROİŞLEMCİ SİSTEMLERİ

Yrd. Doç. Dr. Şule Gündüz Öğüdücü  
<http://www3.itu.edu.tr/~sgunduz/courses/mikroisl/>

1

## Giriş

- Bilgisayar teknolojisindeki gelişme
  - Elektronik öncesi kuşak
  - Elektronik kuşak
  - Mikroişlemci kuşağı

<http://www3.itu.edu.tr/~sgunduz/courses/mikroisl/>

2

## Bilgisayar Tarihi

- Sayıları işleyen ilk aygıt olarak kabul edilen ve milattan önce 1000'li yıllarda Doğu uygarlıklarında kullanılmış olan **abaküs** bilgisayarın atası sayılır. Aygıt, üzerinde boncuklar dizili tellerin bulunduğu tahta bir çerçeveden oluşmaktadır. Abaküs, milattan sonra 1500'lü yıllara kadar kullanılmıştır.



<http://www3.itu.edu.tr/~sgunduz/courses/mikroisl/>

3

## Elektronik Öncesi Kuşak

- İlk Hesap Makinesi: 1642 yılında Blaise Pascal
- Dört işlem ve karekök alma: 1673 yılında Gottfried Leibnitz
- Delgi kartlarına girilmiş desenleri ören dokuma tezgahı: 1804 Joseph Marie Jackard
- Difference Engine: 1822 Charles Babbage
- Analitik Motor: 1835 Charles Babbage
- Sayım Makinesi: 1890 Herman Hollerith
- Turing Hesap Makinesi: 1938 Alan M. Turing

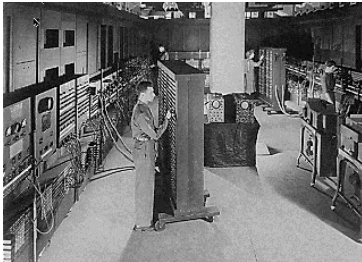


<http://www3.itu.edu.tr/~sgunduz/courses/mikroisl/>

4

## Elektronik Kuşak

- Genel Amaçlı Kullanılan Elektronik İlk Bilgisayar: 1945 yılında ENIAC



<http://www3.itu.edu.tr/~sgunduz/courses/mikroisl/>

5

## Elektronik Kuşak

- Programı Bellekte Saklanan İlk Bilgisayar: John von Neumann tarafından 1952 yılında EDVAC tasarlandı
  - Komutlar ve veriler sayılar şeklinde bir bellekte saklanır
  - Komutları ve verileri saklamak üzere aynı donanım kullanılır
  - Bir komuta bellekteki adresi ile ulaşılır
  - Aynı donanım farklı programları yürütmek için kullanılır
    - genel amaçlı bilgisayar
  - Aynı bellekte farklı programlar bulunabilir

<http://www3.itu.edu.tr/~sgunduz/courses/mikroisl/>

6

## Elektronik Kuşak

- Birinci Nesil Bilgisayarlar (1945-1955): Vakumlu tüpler, veri ve komutları okumak için delgi kartları
- İkinci Nesil Bilgisayarlar (1956-1965): 1948 yılında bulunan transistör bilgisayar teknolojisinde kullanılmıştır.
- Üçüncü Nesil Bilgisayarlar (1966-1975): Çok sayıda transistörlerin üzerinde bulunduğu tümdevreler kullanılmıştır.

<http://www3.itu.edu.tr/~sgunduz/courses/mikroisl/>

7

## Mikroişlemci Kuşağı

- Dördüncü Nesil Bilgisayarlar (1976- ): MİB'nin tek bir tümdevre üzerinde bulunan bilgisayarlar üretildi.
- Gelecek?

<http://www3.itu.edu.tr/~sgunduz/courses/mikroisl/>

8



<http://www3.itu.edu.tr/~sgunduz/courses/mikroisl/>

9

## Mikroişlemcilerin Uygulama Alanları

- Bilgisayar verilen verileri, belirlenen bir programa göre işleyen, istenildiğinde saklayabilen, gerektiği zaman geriye verebilen sayısal bir alettir.
  - Atanmış bilgisayarlar:Yalnızca üretim amaçlarına uygun kullanılırlar.
  - Genel amaçlı bilgisayarlar: Standart bir donanım ile, bellekteki programlar doğrultusunda genel amaçlı olarak kullanılırlar.

<http://www3.itu.edu.tr/~sgunduz/courses/mikroisl/>

10

## Mikroişlemci

- Bilgisayarın temel birimi
  - Aktarılan sayılar üzerinde bellekteki bir program doğrultusunda aritmetik ve lojik işlemler yürütür
  - Sonuç olarak yine sayılar üretir

<http://www3.itu.edu.tr/~sgunduz/courses/mikroisl/>

11

## Mikroişlemci (Devam)

- Programlanabilir: Programda verilen buyruklara göre farklı komutlar yürütebilir.
- Buyruklar: Her mikroişlemci belli işlemleri gerçeklemek üzere tasarlanmıştır. Bu işlemleri gerçeklemek için kendi buyruk kümesi vardır.
- Sayılar: Sadece ikili sayılar üzerinde işlem yapılır. İkili sayıların herbir rakamı bit olarak adlandırılır. Mikroişlemcinin bir anda işleyebileceği bit sayısı sözcük uzunluğunu verir.

<http://www3.itu.edu.tr/~sgunduz/courses/mikroisl/>

12

## Mikroişlemci (Devam)

- Program: Veriyi mikroişlemciye aktaran, işleyen ve sonucu üreten bir dizi buyruk
  - Makine dili: İşlemci mimarisine göre değişen en alt seviye programlama dili. Her buyruk ikili düzende bit kümelerinden oluşur. Mikroişlemci sadece bu buyruklar ile işlem yapar.
    - Örnek: 11001101 anlamı 1+2 olabilir
  - Çevirme dili: (Assembly Language) Makine kodları yerine semboller kullanılır.
  - Yüksek seviye programlama dilleri: C, Fortran, Pascal gibi programlama dilleri

<http://www3.itu.edu.tr/~sgunduz/courses/mikroişlemci/>

13

## Sayı Düzenleri

- Sıkça kullanılan sayı düzenleri:
  - Onluk sayı düzeni: Her rakam için 10 farklı sembol kullanılır: 0,1,2,3,4,5,6,7,8,9
  - İkili sayı düzeni: Taban ikidir ve her rakam 0 veya 1 ile gösterilir.
  - Sekizlik sayı düzeni: Taban sekizdir her basamak için kullanılan semboller şunlardır: 0,1,2,3,4,5,6,7.
  - Onaltılık sayı düzeni: Onaltılık sayı düzeninin tabanı onaltıdır ve kullanılan semboller şunlardır: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

Sayı Düzeni	Taban	Rakamlar
İkili	2	0,1
Sekizlik	8	0,1,2,3,4,5,6,7
Onluk	10	0,1,2,3,4,5,6,7,8,9
Onaltılık	16	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

<http://www3.itu.edu.tr/~sgunduz/courses/mikroişlemci/>

14

## Sayı Düzenleri

- Günlük hayatta kullandığımız onluk sayı düzeninden başka sıkça kullandığımız sayı düzenleri şunlardır:
  - İkili sayı düzeni
  - Sekizlik sayı düzeni
  - Onaltılık sayı düzeni
- Sayısal veriler ikili sayı düzeni ile gösterilir. İkili sayı düzeninde taban ikidir ve her bit 0 veya 1 ile gösterilir.

<http://www3.itu.edu.tr/~sgunduz/courses/mikroişlemci/>

15

## Sayı Dönüşümleri

- Sayıların onluk sayı düzenindeki değerleri her basamaktaki sayının o basamağın ağırlık katsayısıyla çarpılması ve bulunan sayıların toplanması ile elde edilir. Her basamağın ağırlık katsayısı tabanın kuvveti alınarak bulunur, basamağın konumu kaçınıcı kuvveti olacağını belirler.

$$953,78 = 9 \cdot 10^2 + 5 \cdot 10^1 + 3 \cdot 10^0 + 7 \cdot 10^{-1} + 8 \cdot 10^{-2} \\ = 900 + 50 + 3 + 0,7 + 0,08 = 953,78$$

$$\%1011 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 \\ = 8 + 0 + 2 + 1 = 11$$

$$\$A2F = 10 \cdot 16^2 + 2 \cdot 16^1 + 15 \cdot 16^0 \\ = 2560 + 32 + 15 = 2607$$

- % sembolü ikili sayılar için, \$ veya h ise 16'lık sayılar için kullanılır, örneğin \$A2F veya A2Fh.

<http://www3.itu.edu.tr/~sgunduz/courses/mikroişlemci/>

16

## Sayı Dönüşümleri

- Onluk İkili dönüşümü:
  - Arama Yöntemi: Sayının içinde ikinin kuvvetlerini aramaya dayanır.

Onluk sayı	En büyük $2^n$	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
169	128	1	0	1	0	1	0	0	1
41	32			1					
9	8				1				
1	1						1		

<http://www3.itu.edu.tr/~sgunduz/courses/mikroişlemci/>

17

## Sayı Dönüşümleri

- Onluk - İkili Dönüşümü:
  - Bölme Yöntemi: Onluk sayı ikiye bölünür. Bölme sonucunda kalanın 1 veya 0 olmasına bakılarak ikili tabandaki karşılığı bulunur.

Onluk sayı	kalan	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
169	169/2=84	1							
84	84/2=42	0							
42	42/2=21	0							
21	21/2=10	1							
10	10/2=5	0							
5	5/2=2	1							
2	2/2=1	0							
1	1/2=0	1							

<http://www3.itu.edu.tr/~sgunduz/courses/mikroişlemci/>

18

## Sayı Dönüşümleri

- İkili - Sekizlik Dönüşümü:  
 $10101001_2 = (010\ 101\ 001)_2 = 251_8$
- İkili - Onaltılık dönüşüm:  
 $10101001_2 = (1010\ 1001)_2 = A9h$
- İkili - Onluk sayılar (Binary Coded Decimal):  
Onluk düzendeki her basamak ikili-onluk kodda 4 bit ile kodlanır.  
 $13_{10} = 1101_2 = 0001\ 0011$

<http://www3.itu.edu.tr/~sgunduz/courses/mikroisl/>

19

## Sayı Dönüşümleri

İkili sayı	İkili Sayı	Sekizlik Sayı	Onaltılık Sayı	İkili-Onluk Sayı
0	0000 0000	000	00	0000
1	0000 0001	001	01	0001
2	0000 0010	002	02	0010
3	0000 0011	003	03	0011
4	0000 0100	004	04	0100
5	0000 0101	005	05	0101
6	0000 0110	006	06	0110
7	0000 0111	007	07	0111
8	0000 1000	010	08	1000
9	0000 1001	011	09	1001
10	0000 1010	012	0A	0001 0000
11	0000 1011	013	0B	0001 0001
12	0000 1100	014	0C	0001 0010
13	0000 1101	015	0D	0001 0011
14	0000 1110	016	0E	0001 0100
15	0000 1111	017	0F	0001 0101
16	0001 0000	020	10	0001 0110
20	0001 0100	024	14	0010 0000
160	1010 0000	240	AD	0001 0110 0000
248	1111 1000	370	F8	0010 0100 1000

<http://www3.itu.edu.tr/~sgunduz/courses/mikroisl/>

20

## Bilgisayarda Verilerin Gösterilmesi

- Bilgisayar belleğinin ve merkezi işlem biriminin yapısı nedeniyle veriler bellekte belli kalıplar içinde saklanırlar. Bilgisayarlar için verilerin ikili düzende saklanması uygundur.
- Sayıların hangi aralıklarda değişebileceği kullanılan bit sayısına bağlıdır. Saklama ortamları için genel olarak kullanılan birim sekiz bittir (*1 byte*).

<http://www3.itu.edu.tr/~sgunduz/courses/mikroisl/>

21

## Bilgisayarda Verilerin Gösterilmesi

- Bilgisayarda işaretli ve işaretli sayıları nasıl gösterebiliriz?
- Bilgisayarda büyük sayıları nasıl oluşturabiliriz?
- Tam ve ondalıklı sayıları nasıl gösterebiliriz?

<http://www3.itu.edu.tr/~sgunduz/courses/mikroisl/>

22

## İşaretsiz Tam Sayılar

- Sekiz bitlik bir belleğin herbir gözü bir sayı saklamak amacıyla kullanılırsa  $2^8=256$  değişik işaretsiz sayıyı saklama olanağı vardır.

0000 0000 0  
1111 1111 255

<http://www3.itu.edu.tr/~sgunduz/courses/mikroisl/>

23

## İşaretili Tam Sayılar

- İkili bir sayının işaretini belirtmek için iki yöntem kullanılır:
  - İkili sayıdaki sekiz bittten bir tanesini işaret bitine ayrılır. En Yüksek Anlamlı Bit (YAB) işaret biti olarak kullanılır. YAB'in 1 olması sayının negatif, 0 olması pozitif olduğu anlamına gelir.

1111 1111 -127  
1000 0001 -1  
0000 0000 0  
0000 0001 1  
0111 1111 127

- Tümlenme yöntemi: Bu yöntemde pozitif bir sayının negatif karşılığı sayının 2'ye tümlenmesi alınarak bulunur. İkili bir sayının 2'ye tümlenmesini bulmak için önce sayının 1'e tümlenmesi alınır ardından 1 eklenir.
  - 1'e tümlenmesini al: Sayı içindeki 0'lar yerine 1, 1'ler yerine 0 koy.
  - 1 ekle.

1000 0000 -128  
1111 1111 -1  
0000 0000 0  
0000 0001 1  
0111 1111 127

<http://www3.itu.edu.tr/~sgunduz/courses/mikroisl/>

24

## İşaretili Tam Sayılar

- İşaretili sayıların gösteriliminde kullanılan her iki yöntemde de ortak olan nokta negatif sayılar için her ikisinde de YAB'in 1 olmasıdır.

Örnek: +7                      0000 0111

- İşaretili Sayılar: -7 -> 1000 0111

- Tümleyen Sayılar:

1'e tümleyeni: 1111 1000

2'ye tümleyeni: 1111 1001 -> -7

<http://www3.itu.edu.tr/~sgunduz/courses/mikroisl/>

25

## Bilgisayarda Büyük Sayıların Kullanımı

- Daha büyük değerlerde tam sayıların bilgisayar belleğinde saklanabilmesi için birden fazla bellek gözü kullanılabilir. İki sekiz bitlik bellek gözü onaltı bitlik bir sayı saklamak için kullanılabilir. Bu şekilde sayının yarısı bir bellek gözünde, diğer yarısı diğer bellek gözündedir.

- 16 bit içine işaretsiz olarak 0 ile  $2^{16}=65535$  arası sayılar yazılabilir.

- İşaretili olarak -32767 ve 32767 arası sayılar veya tümleyen sayı biçiminde -32768 ve 32767 arası sayılar yazılabilir.

- Bilgisayar aritmetiğinde tam sayıların boyu en uygun biçimde kullanmak amacıyla program başında belirlenebilir.

<http://www3.itu.edu.tr/~sgunduz/courses/mikroisl/>

26

## Bilgisayarda Ondaklı Sayıların Gösterilmesi

- Ondaklı sayı önce yasal biçime dönüştürülür. Bunun için sayıdaki virgöl sayının en soluna getirilir ve sayının değer kaybını gideren bir değer ile çarpılır.

Asıl sayı                      205,03

Yasal Yazılımı               $0,20503 \times 10^3$

Sayının bu şekilde yazılması ile ortaya çıkan iki parçaya yalın sayı ve üs adı verilmektedir.

Yalın Sayı                      20503

Üs                                      3

Yalın sayı 16 bitlik ve üs 8 bitlik bir alana yazılabilir. Yalın sayı ve üs işaretili sayı veya tümleyen sayı biçiminde gösterilebilir.

Üssün işareti	üs	Yalın işareti	Yalın sayı
0	000 0011	0	101 0000 0001 0111

<http://www3.itu.edu.tr/~sgunduz/courses/mikroisl/>

27

## İşaretsiz Tam Sayılarda Toplama İşlemi

- Toplama: İkili sayıların toplanmasında aşağıdaki kurallar geçerlidir:

$0 + 0 = 0$   
 $0 + 1 = 1$   
 $1 + 0 = 1$   
 $1 + 1 = 0$       elde 1

158: 1001 1110

47: +0010 1111

205: 1100 1101

254: 1111 1110

2: + 0000 0010

256: 1 0000 0000

**ELDE**

- İki n bitlik işaretsiz sayının toplanması sonunda (n+1) bitlik bir sayı oluşursa elde oluşmuştur denir.

<http://www3.itu.edu.tr/~sgunduz/courses/mikroisl/>

28

## İşaretsiz Tam Sayılarda Çıkarma İşlemi

- İkili sayı düzeninde çıkarma işlemi onluk sayı düzeninde olduğu gibi gerçekleşir. Ancak bilgisayarlarda kullanılan yöntem tümleyen aritmetiğine göre çıkarmadır. Bu yöntemde ana sayı ile çıkartılan sayının 2'ye tümleyeni toplanarak sonuç elde edilir.

5: 0000 0101                      0000 0101  
2: 0000 0010                      + 1111 1110  
3:                      2'ye tümleyeni      1 0000 0011                      **ELDE=BORÇ**

2: 0000 0010                      0000 0101  
5: 0000 0101                      + 1111 1011  
-3:                      2'ye tümleyeni      1111 1101                      **ELDE YOK - BORÇ OLUŞTU**

- İşaretsiz sayıların çıkartılmasında çıkartılan sayı çıkarılan sayıdan büyükse borç oluşur. İşaretsiz sayıların çıkarılmasında tümleyen aritmetiğine göre toplama yapıldığında elde biti oluşursa borç yok demektir.

<http://www3.itu.edu.tr/~sgunduz/courses/mikroisl/>

29

## İşaretili Tam Sayılarda Toplama İşlemi

- İşaretsiz tam sayılarda olduğu gibi yapılır.

-1: 1111 1111

-1: + 1111 1111

-2: 1 1111 1110

+127: 0111 1111

+ 1: +0000 0001

+128: 1000 0000

- İki pozitif sayının toplanması sonucunda negatif sayı oluştu. **Taşma** vardır.

<http://www3.itu.edu.tr/~sgunduz/courses/mikroisl/>

30

## İşaretili Tam Sayılarda Çıkarma İşlemi

- İşaretsiz tamsayılarda olduğu gibi ikiye tümleyen yöntemi ile yapılır.
- İşaretili tamsayıların çıkarılması işleminde de taşma oluşabilir.

<http://www3.itu.edu.tr/~sgunduz/courses/mikroisi/>

31

## Taşma Oluşması

- İşaret bitine gelen ve işaret bitinden çıkan eldeye bakılarak taşmanın oluşması kontrol edilebilir. Bu elderler farklıysa taşma oluşur, elderler aynıysa taşma oluşmaz.

Eldesiz	01	Eldesiz	10
70:	0 100 0110	-70:	1 011 1010
80:	+ 0 101 0000	-80:	+ 1 011 0000
150:	1 001 0110	-150:	0 110 1010

<http://www3.itu.edu.tr/~sgunduz/courses/mikroisi/>

32

## İkilik Sayılar Üzerinde Aritmetik İşlemler

- **Eldesiz:** İki n bitlik işaretsiz sayının toplanması sonunda (n+1) bitlik bir sayı oluşursa elde oluşmuştur denir.
- **Borç:** İşaretsiz sayıların çıkartılmasında çıkartılan sayı çıkarılan sayıdan büyükse borç oluşur. İşaretsiz sayıların çıkarılmasında tümleyen aritmetiğine göre toplama yapıldığında elde biti oluşursa borç yok demektir.
- **Taşma:** n bitlik işaretili sayıların toplanması veya çıkartılması sonunda (n+1) bitlik bir sayı oluşursa taşma vardır denir. Şu durumlarda taşma oluşabilir:

poz + poz → neg	poz - neg → neg
neg + neg → poz	neg - poz → poz

<http://www3.itu.edu.tr/~sgunduz/courses/mikroisi/>

33