

BİL 362 Mikroişlemciler: Giriş

Ahmet Burak Can
abc@hacettepe.edu.tr

Ders İçeriği

- Mikroişlemcilere giriş
 - Bilgisayar Sisteminin Tanımı
 - Makine Dilleri, Çevirici (Assembly) Dilleri
 - Sayı Sistemleri
 - Veri Gösterim Yöntemleri
 - Temel Bilgisayar Mimarisi
- Intel 8086 Mimarisi
 - Veri Türleri ve Özellikleri
 - Yazmaçlar
- Adresleme Modları

Ders İçeriği

- Veri Aktarım Komutları
 - Yazmaçtan yazmaca aktarım
 - Yazmaç - Ana Bellek arası aktarım
 - Yığıt işlemleri
- Aritmetik İşlem Komutları
- Akış Denetim Komutları
 - Koşulsuz Atlama Komutları
 - Koşullu Atlama Komutları
 - Yordam Çağırma Komutları
- İşlemci Denetim Komutları

Ders İçeriği

- Dizgi İşlemleri
- Kesilmeler
 - Ekran kesilmeleri
 - Klavye kesilmeleri
 - Seri Kapı (serial port) kesilmeleri

Ders Kaynakları

- The Intel Microprocessors Eighth Edition
Barry B. Brey, Pearson Education
- Ders notları

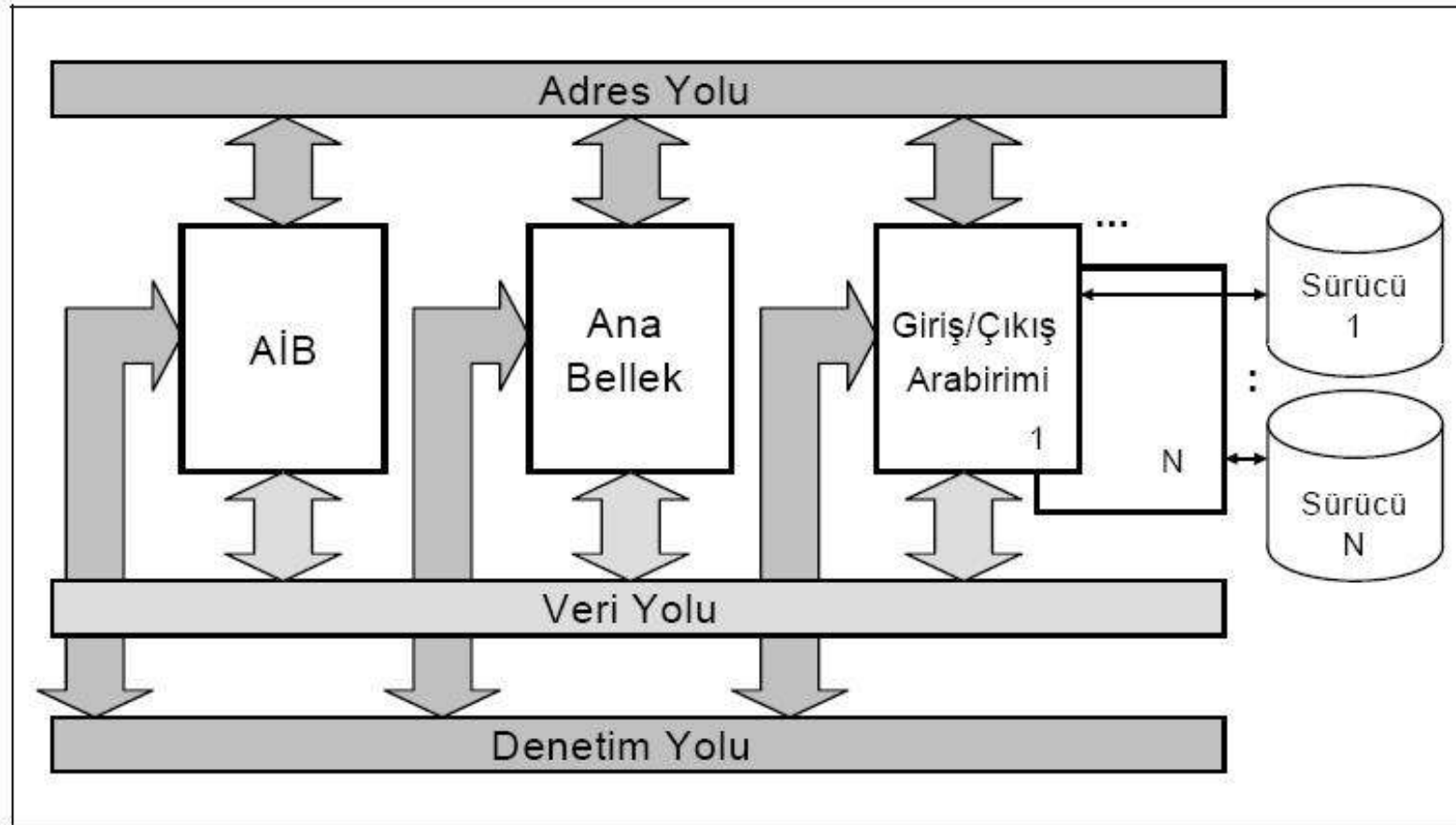
Ders Değerlendirmesi

- Arasınava (% 40)
- Final sınavı (% 60)
- Ödevler (%15)
 - Ödev puanı, eğer sınavlardan toplam 40 puan ve üzeri alırsanız geçerli olacaktır.
- Notlandırmada üniversite katalog uygulaması geçerli olacaktır.
- Yönetmelik gereği, derslerin en az % 70'ine katılmanız mecburidir.

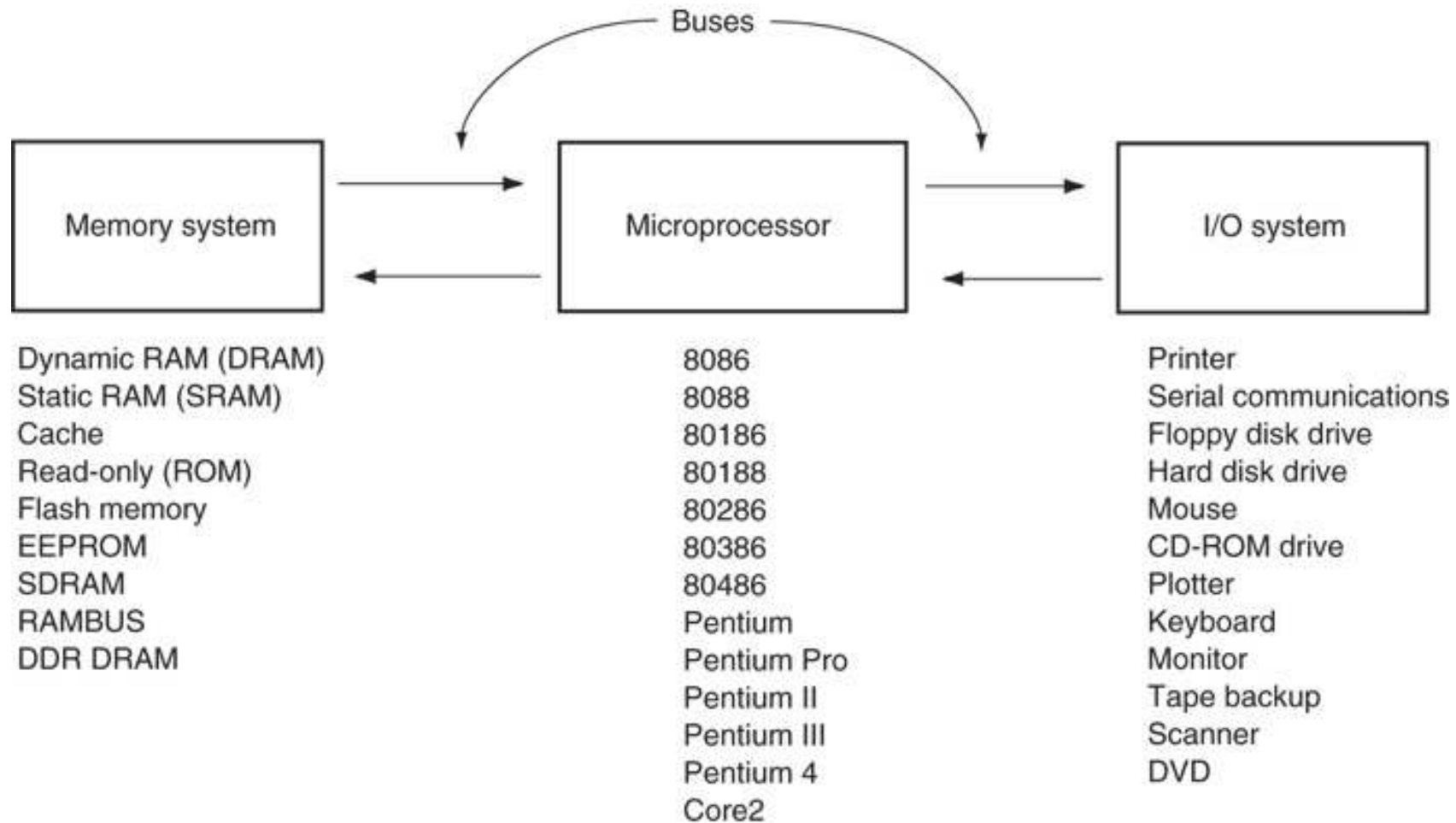
Bilgisayar Sistemi

- Bilgisayar, aritmetik ya da mantık işlemlerini önceden yüklenmiş programlar sayesinde çözebilen bir araçtır.
- Von Neumann tanımına göre bilgisayarın bileşenleri:
 - Merkezi İşlem Birimi
 - Ana Bellek
 - Giriş/Çıkış Birimleri

Bilgisayarın Temel Yapısı



Bazı Temel Bilgisayar Teknolojileri



Bilgisayarın Bazı Önemli Bileşenleri

- Merkezi İşlem Birimi
 - Yazmaçlar, Cache Bellek
- Ana kart
 - Bus, BIOS
- Ana Bellek (RAM)
- Ekran Kartı
- Sabit Disk
- Klavye
- Fare
- Yazıcı

Bilgisayarın Tarihçesi

- 1642’de Blaise Pascal tarafından hesap makinesi üretildi
- 1820’de Xavier Thomas tarafından ticari amaçlı ilk hesap makinesi üretildi.
- 1890’da Herman Hollerich tarafından bulunan delikli kart teknolojisini, yaklaşık 80 yıl kullanıldı.
- 1937’de Amerikan Howard Hathaway Aiken’in yönettiği bir araştırma takımı, delikli kart kullanan Mark–1’i üretti.
- 1945’de, 2. Dünya Savaşı’nın bitmek üzere olduğu bir zamanda, askeri amaçla ENIAC üretildi.
 - Bir futbol sahası alan kaplıyordu.
 - Cihazın içine yuva yapan böcekleri temizlemek işleminden dolayı **debug** terimi ortaya çıktı.

Bilgisayarın Tarihçesi

- 1951'de Amerikalılar tarafından üretilen UNIVAC ilk ticari bilgisayar oldu.
- IBM firması 1958'den itibaren bilgisayarda vakum tüpleri yerine diyot ve transistörleri kullanmaya başladı.
- 1964 yılından itibaren bütünleşik devre (IC – Integrated Circuit) teknolojisi kullanılmaya başlandı.

Makina Dilleri

- Merkezi işlem biriminde çalıştırılabilecek her komutun bir sayısal kodu vardır.
- Makine dilleri 0 ve 1'lerden oluşan en alt seviyeli dillerdir.
 - Bu dilde, makina komutları bir sayısal değer olarak kodlanır.
- Bu diller, işlemci ve donanıma bağlı olarak değişmektedir.
- Her işlemci mimarisinin kendine özgü bir makina dili vardır.
- Makina dili düzeyinde programlama yapılmamaktadır.

Çevirici Diller

- Makina dili komutlarını temsil eden sözde komutların tanımlandığı dillerdir.
- İşlemcinin çalıştırabileceği her makina komutu için bir çevirici komutu tanımlanır.
- Her işlemci platformu için farklı çevirici dilleri mevcuttur.
- Örneğin, bir Motorola işlemci ile Intel işlemcinin çevirici dilleri birbirinden farklıdır.

Çevirici Dillere Örnekler

x86 ASM (İşletim Systemsiz)

```
MOV      AH,03H
INT 10H
MOV      AL,01H
MOV      BH,00H
MOV      BL,01001111B
MOV      CX,MESAJ_SON-MESAJ_BAS
PUSH     CS
POP      ES
```

Çevirici Dillere Örnekler

x86 ASM (İşletim Systemsiz) - DEVAMI

```
MOV      BP, MESAJ_BAS
MOV      AH,13H
INT 10H
JMP      SON
MESAJ_BAS DB      'Merhaba Dünya!'
MESAJ_SON DB      00H
SON:     RET
```


Çevirici Dillere Örnekler

DOS:

```
MOV     AX,CS
MOV     DS,AX
MOV     AH,09H
MOV     DX,OFFSET MESAJ
INT 21H
XOR     AX,AX
INT 21H
MESAJ:
DB  "Merhaba Dunya!",13,10,"$"
```

Çevirici Dillere Örnekler

WINDOWS:

```
TITLE Merhaba Dunya ASM. Tasm
VERSION T310
Model use32 Flat,StdCall
start_code segment byte public 'code' use32
Begin:
    Call MessageBox,0,offset sMesaj, offset sBaslik,0
    Call ExitProcess,0
start_code Ends
start_data segment byte public 'data' use 32
sMesaj DB 'Merhaba Dunya',0
sBaslik DB "Merhaba!",0
start_data Ends
End begin
```

Çevirici Dillere Örnekler

LINUX:

```
SECTION .data
mesaj      db "Merhaba Dünya!",0xa;
uzunluk     equ $ - mesaj
SECTION .text
global main
main:       mov     eax,4
            mov     ebx,1
            mov     ecx,mesaj
            mov     edx,uzunluk
            int     0x80
            mov     eax,1
            mov     ebx,0
            int     0x80
```

Çevirici Dillere Örnekler

68000 Amiga ASM

```
move.l      #DOS
move.l      4.w,a6
jsr  -$0198(a6)
move.l      d0,a6
beq.s       .Out
move.l      #MerhabaDunya,d1
jsr          -$03B4
move.l      a6,a1
move.l      4.w,a6
jsr          -$019E(a6)
.Outrts
DOS          dc.b 'dos.library',0
MerhabaDunya dc.b 'Merhaba Dunya!',$A,0
```

Üst Düzey Programlama Dilleri

- Çevirici dillerle büyük programların gerçekleştirimi ve bakımı zor olduğu için daha üst düzey programlama dilleri tanımlanmıştır. Bazı örnekleri:
- C, C++, Java, C#, ADA
- Fortran, Pascal, COBOL, BASIC

Sayı Sistemleri

- 1'lik sayı sistemi
- Roma sayı sistemi
- Hint-Arap sayı sistemi
- 10'luk sayı sistemi
- 2'lik sayı sistemi
- 8'lik sayı sistemi
- 16'lik sayı sistemi

1'lik Sayı Sistemi



IIIIII Elma



I Şeftali

Roma Sayı Sistemi

Romen Rakamları	Büüklük
I	1
V	5
X	10
L	50
C	100
D	500
M	1000

- MCMLXXXVI \rightarrow M – CM – LXXX – VI \rightarrow 1000 + 900 + 80 + 6 = 1986
- MMVI \rightarrow M – M – VII \rightarrow 1000 + 1000 + 7 = 2007

Hint-Arap Sayı Sistemi

- Romen sayı sisteminde sadece toplama ve çıkarma yapılabilmektedir. Ayrıca basamak kavramı yoktur.
- Bu sorunlar yüzünden bugün kullanılan rakamlar ve sayı sistemi geliştirilmiştir.
- Günümüzde kullanılan 0, 1, 2, 3, 4, 5, 6, 7, 8 ve 9 rakamları Hint-Arap rakamlarıdır.
- 10'luk sistemin temelini bu sayı sistemi oluşturur.

Bilgisayarda Rakamlar

- Sayısal bir sistemdeki yazılımlar aslında bir rakamlar topluluğudur.
- Yazılımları oluşturan verilerin ve komutların gerçek elektrik devrelerinde gerilim ve akım denetimi kullanılarak yaratılır.
- Günümüzde bütün sayısal bilgisayarlarda yüksek ve düşük olmak üzere iki çeşit sinyal kullanılmaktadır.
- İkili düzendeki sayılar, 1 (yüksek) ve 0 (düşük) değerlerinden oluşur.
- Bu düzendeki sayıların 0 ve 1'den oluşan her bir basamağına "bit" denir.
- Bilgisayarlar, yalnızca 0 ve 1 rakamlarını kullandıklarından ikili tabana göre çalışırlar.

İkilik Taban

- Bütün sayısal değerlerin sadece 0 ve 1 rakamları ile ifade edildiği sayı sistemidir.

Rakam	İkilik Tabandaki Kod
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Örnek sayılar:

$(10011)_2$

$(111)_2$

$(101111)_2$

$(101001)_2$

$(11111011011)_2$

$(01110011)_2$

$(10011111)_2$

İkilik - Onluk Taban Çevrimleri

- Onluk sistemde 12345 sayısı basamaklarına şu şekilde ayrıştırılır:

$$(12345)_{10} = 5 \times 10^0 + 4 \times 10^1 + 3 \times 10^2 + 2 \times 10^3 + 1 \times 10^4$$

- $(11000000111001)_2$ sayısı onluk sisteme çevrilirken, yukarıdakine benzer şekilde basamaklarına ayrıştırılabilir:

$$(11000000111001)_2 = 1 \times 2^0 + 0 \times 2^1 + 0 \times 2^2 + 1 \times 2^3 + 1 \times 2^4 + 1 \times 2^5 + 0 \times 2^6 + 0 \times 2^7 + 0 \times 2^8 + 0 \times 2^9 + 0 \times 2^{10} + 0 \times 2^{11} + 1 \times 2^{12} + 1 \times 2^{13}$$

$$(11000000111001)_2 = 1 + 0 + 0 + 8 + 16 + 32 + 0 + 0 + 0 + 0 + 0 + 0 + 4096 + 8192$$

$$(11000000111001)_2 = 12345$$

Onluk - İkili Taban Çevrimleri

- Onluk tabandan ikili tabana geçerken, onluk sayıya art arda bölme işlemi uygulanır.
- Bölmenin sonucunda ortaya çıkan kalan değerler ikili sayının basamakları olarak alınır.
- Bölme sonucu oluşan bölüm değeri 2'den küçük olana kadar bölme sürdürülür.

Onluk - İkili Taban Çevrimleri

- $(123)_{10}$ sayısını onluk sisteme çevirelim.

$$123 / 2 = 61 \quad \text{Kalan}=1 \quad (s_0 = 1)$$

$$61 / 2 = 30 \quad \text{Kalan}=1 \quad (s_1 = 1)$$

$$30 / 2 = 15 \quad \text{Kalan}=0 \quad (s_2 = 0)$$

$$15 / 2 = 7 \quad \text{Kalan}=1 \quad (s_3 = 1)$$

$$7 / 2 = 3 \quad \text{Kalan}=1 \quad (s_4 = 1)$$

$$3 / 2 = 1 \quad \text{Kalan}=1 \quad (s_5 = 1)$$

$$1 \quad (s_6 = 1)$$

$$(s_6 s_5 s_4 s_3 s_2 s_1 s_0)_2 = (1111011)_2$$

Onaltılık Taban

- Onaltılık tabanda ise sayıların gösterilmesi için 16 ayrı rakam gerektiğinden, 9'dan sonraki rakamları göstermek için harflerden yararlanılır.

Onaltılık Rakam	Onluk Değer
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	10
B	11
C	12
D	13
E	14
F	15

Onaltılık – Onluk Taban Çevrimleri

- Onaltılık tabandaki sayılar onluk tabana çevrilirken basamaklarına ayırma yöntemi uygulanır.

$$(A12F)_{16} = 15 \times 16^0 + 2 \times 16^1 + 1 \times 16^2 + 10 \times 16^3$$

$$(A12F)_{16} = 15 \times 1 + 2 \times 16 + 1 \times 256 + 10 \times 4096$$

$$(A12F)_{16} = 15 + 32 + 256 + 40960$$

$$(A12F)_{16} = 41263$$

Onluk - Onaltılık Taban Çevrimleri

- Onluk tabandaki sayılar onaltılık tabana çevrilirken onluk düzendeki sayı, bölüm 16'dan küçük oluncaya dek 16'ya bölünür. Örnek:

$$41263 / 16 = 2578$$

$$\text{kalan} = 15$$

$$s_0 = 15 = F$$

$$2578 / 16 = 161$$

$$\text{kalan} = 2$$

$$s_1 = 2$$

$$161 / 16 = 10$$

$$\text{kalan} = 1$$

$$s_2 = 1$$

$$10$$

$$s_3 = 10 = A$$

$$(s_3 s_2 s_1 s_0)_{16} = (A12F)_{16}$$

İkili – Onaltılık Taban Çevrimleri

- İkili düzendeki sayının basamakları sol yandan başlayarak dörderli öbeklere ayrılır.
- Bu öbeklerin onaltılık tabandaki karşılıkları bulunur ve öbekler bulunan onaltılık rakamlarla değiştirilir.
- Böylece ortaya çıkan sayı ikili tabandaki asıl sayının onaltılık tabana dönüştürülmüş biçimidir.

İkili – Onaltılık Taban Çevrimleri

Örnek: $(11111010101010110)_2$ sayısını 16'lık tabana çevirelim.

- Sayıyı dörderli öbeklere bölersek:

$$(1)_2 (1111)_2 (0101)_2 (0101)_2 (0110)_2$$

- Herbir dörtlü öbeğin değerini bulalım:

$$(0001)_2 = 1 = (1)_{16}$$

$$(1111)_2 = 15 = (F)_{16}$$

$$(0101)_2 = 5 = (5)_{16}$$

$$(0101)_2 = 5 = (5)_{16}$$

$$(0110)_2 = 6 = (6)_{16}$$

- Böylece elde edilen 16'lık sayı: $(1F556)_{16}$

Onaltılık – İkili Taban Çevrimleri

- Onaltılık tabandaki bir sayının ikili tabana çevrilmesi için açıklanan yöntemin tersi kullanılabilir.
- Bu durumda onaltılık sayının her bir basamağının ikili tabandaki dört bitlik karşılığı yazılır ve bu değerler birleştirilir.
- **Örnek:** $(ABC)_{16}$ sayısı ikili tabana çevirelim:

$$(C)_{16} = (1100)_2$$

$$(B)_{16} = (1011)_2$$

$$(A)_{16} = (1010)_2$$

Bu değerler birleştirilirse $(101010111100)_2$ sayısı elde edilir.

İkilik Tabanda Tümleyenler

- Tümleyenler, bilgisayarlarda çıkarma işlemini ve mantık işlemlerini yalın bir biçimde gerçekleştirmek amacıyla kullanılır.
- Bilgisayarlarda eksi sayılar, tümleyen yöntemleri kullanılarak saklanır.
- İkilik tabanda kullacağımız iki tümleyen tanımı:
 - 1'e tümleyen
 - 2'ye tümleyen

1'e Tümleyen

- İkili tabanda bir sayının 1'e tümleyenini bulmak için, her bitin (yani ikili sayının her bir basamağının) değeri terslenir.
 - Bit değeri 1 ise, sonuç olarak 0 alınır.
 - Bit değeri 0 ise, sonuç olarak 1 alınır.
- 1'e tümleyen için, devre düzeyinde DEĞİL kapıları kullanılır.

Sayı	1'e tümleyeni
1	0
0	1
101	010
1111	0000
11011010101001111	00100101010110000

2'ye Tümleyen

- İkili tabandaki bir sayının 2'ye tümleyenini bulmak için sayının önce 1'e tümleyeni alınır.
- Bulunan sonuç bir ile toplanır.
- Toplama sonucunda bulunan sayının basamak sayısı, tümleyeni alınan sayının basamak sayısından fazla olursa, sonuçta bulunan sayının en soldaki basamağı atılır.

Sayı	2'ye tümleyeni
1	1
0	0
101	011
1111	0001
1101101	0010011

2'ye Tümleyen İçin Alternatif Yöntem

- Sayının sağından başlayıp soluna doğru ilerleyerek değeri 1 olan ilk basamağa kadar bulunan tüm 0'ları kopyala.
 - Eğer sayının en sağdaki biti bir ise bu adım atlanır.
- 0'ların ardından gelen ilk 1'i kopyala.
- Geri kalan basamakların tamamının tersini al.

İkili Tabanda Toplama İşlemi

- İkili tabanda toplama işlemi, onluk tabandaki toplama işleminin yapıldığı gibi iki sayının her bir basamağının ayrı ayrı toplanması yoluyla yapılır.
- Tabana göre elde var ise (yani iki basamağın toplamı birden büyük çıktıysa) elde bir soldaki basamağa eklenir.
- **Örnek:**

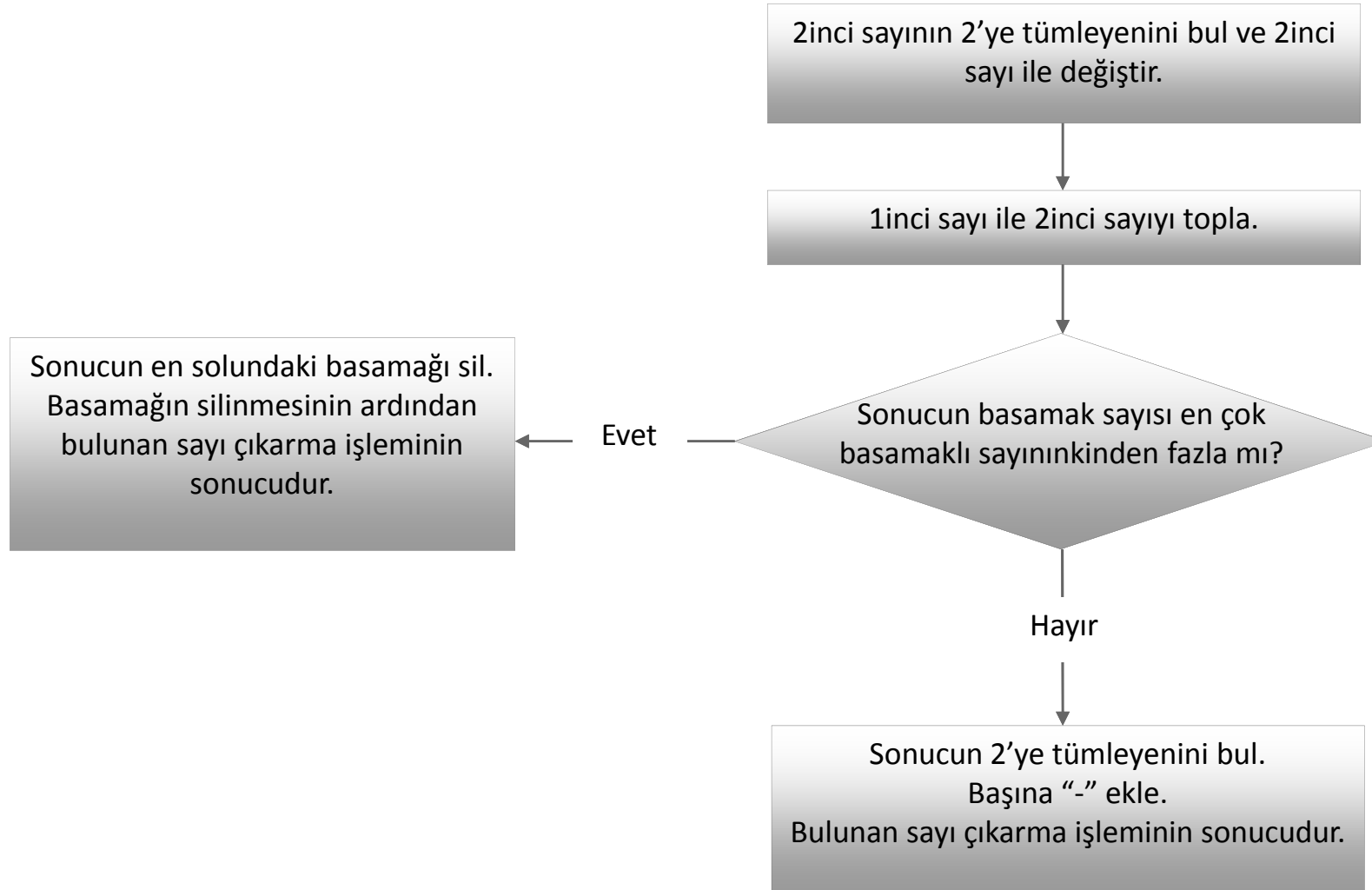
$$\begin{array}{r} 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1 \\ + 0 \ 1 \ 1 \\ \hline 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \end{array}$$

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

İkili Tabanda Çıkarma İşlemi

- Bilgisayarlarda çıkarma işlemi yapılırken tümleyenler kullanılır.
- Bire tümleyen veya ikiye tümleyen kullanılabilir.
- İkiye tümleyen ile çıkarma yapılırken, çıkarılan sayının tümleyeni (bir diğer deyişle eksilisi) alınır ve bulunan bu sayı diğer sayı ile toplanır.
- Eğer toplama sonucu, işlenenlerin basamak sayısından fazla ise, en soldaki bit silinir.
 - Böyle bir durum oluşması, bulunan sonucun sıfırdan büyük olduğunu gösterir.

İkiye Tümleyen Yöntemiyle Çıkarma İşlemi



İkiye Tümleyen Yöntemiyle Çıkarma İşlemi

- **Örnek:**

110001101 – 011110101 işlemini ikiye tümleyen yöntemiyle yapalım.

- 011110101 sayısının ikiye tümleyenini alarak 100001011 değerini elde ederiz.

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

- Toplama sonucu, işlenenlerden büyük olduğu için en soldaki bit atılır. Çıkarma sonucu sıfırdan büyüktür.
- **Sonuç : 010011000**

İkiye Tümleyen Yöntemiyle Çıkarma İşlemi

- **Örnek:**

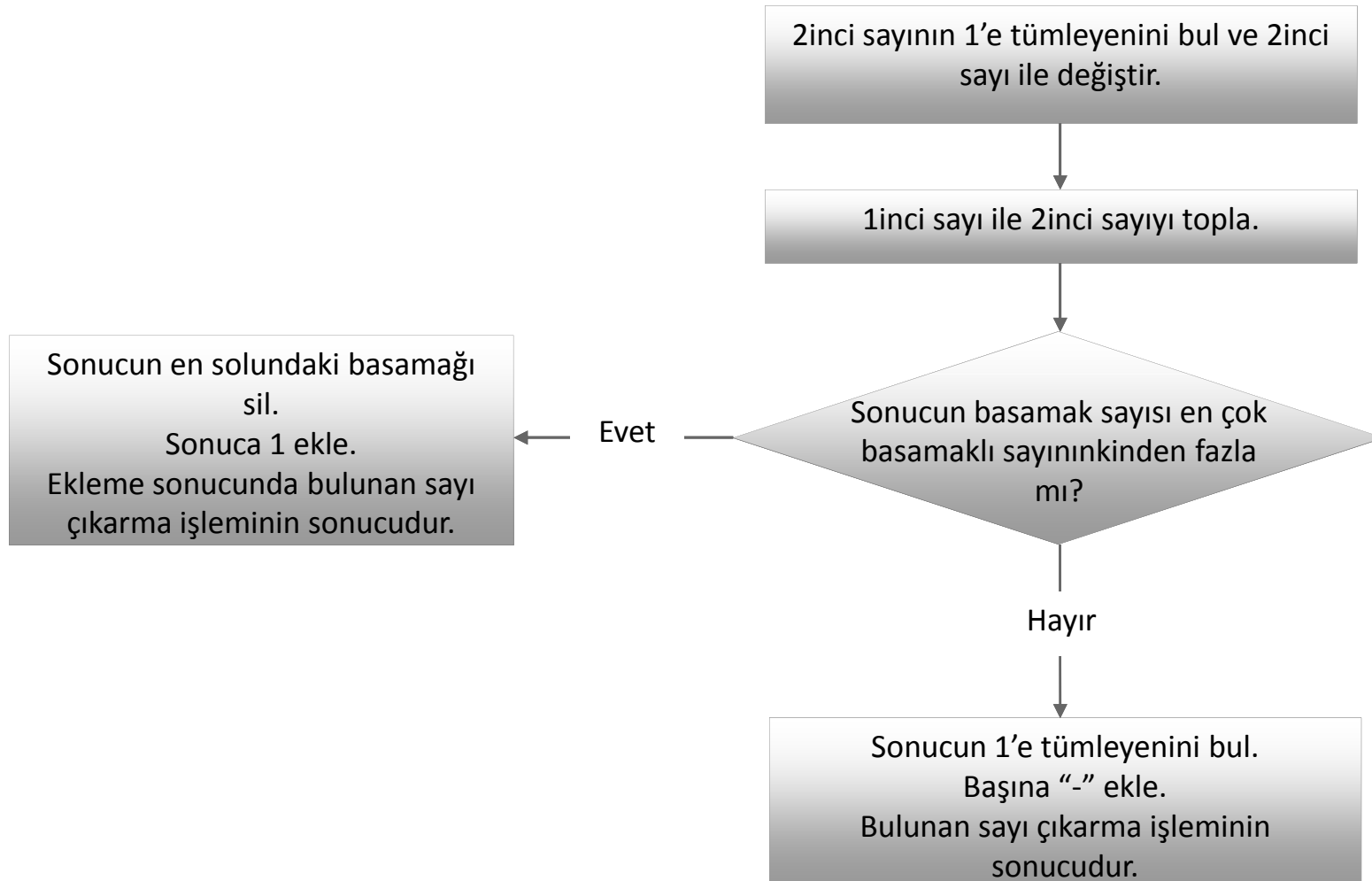
011110101 – 110001101 işlemini ikiye tümleyen yöntemiyle yapalım.

- 110001101 sayısının ikiye tümleyenini alarak 001110011 değerini elde ederiz.

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

- Toplama sonucu, işlenenler ile aynı sayıda basamak içerdiği için çıkarma sonucu sıfırdan küçüktür. Sonucun ikiye tümleyeni alınır ve başına – işareti konur.
- **Sonuç : – 010011000**

Bire Tümleyen Yöntemiyle Çıkarma İşlemi



Bire Tümleyen Yöntemiyle Çıkarma İşlemi

- **Örnek:**

110001101 – 011110101 işlemini bire tümleyen yöntemiyle yapalım.

- 011110101 sayısının bire tümleyenini alarak 100001010 değerini elde ederiz.

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

- Toplama sonucu, işlenenlerden büyük olduğu sonuç sıfırdan büyüktür. Çıkarma sonucunun en soldaki biti atılır ve 1 eklenir.
- **Sonuç : 010011000**

Bire Tümleyen Yöntemiyle Çıkarma İşlemi

- **Örnek:**

011110101 – 110001101 işlemini bire tümleyen yöntemiyle yapalım.

- 110001101 sayısının bire tümleyenini alarak 001110010 değerini elde ederiz.

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

- Toplama sonucu, işlenenler ile aynı sayıda basamak içerdiği için çıkarma sonucu sıfırdan küçüktür. Sonucun bire tümleyeni alınır ve başına – işareti konur.

– **Sonuç : – 010011000**

Onaltılık Tabanda Toplama İşlemi

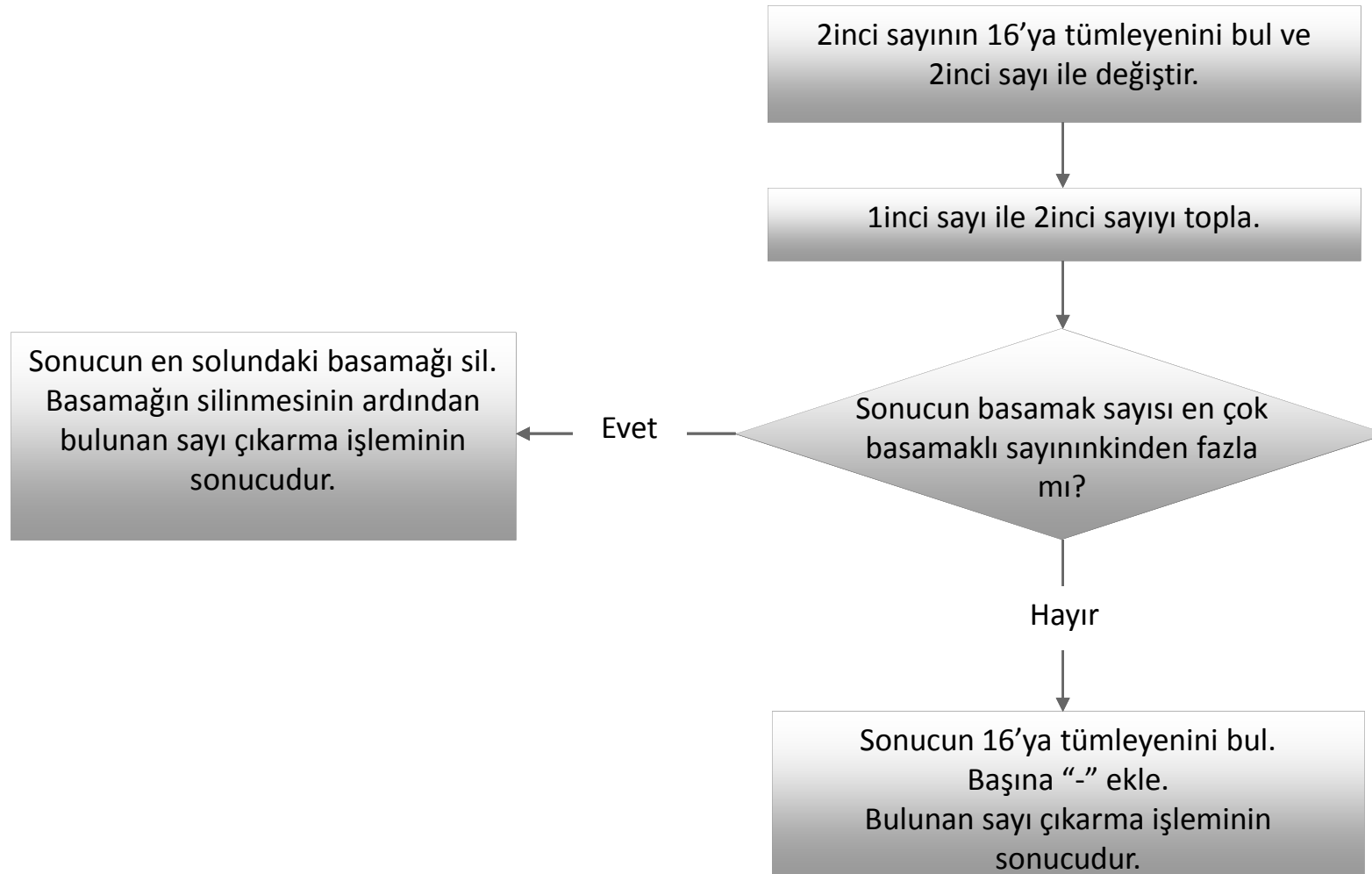
- Diğer tabanlarda olduğu gibi, basamaklar ayrı ayrı toplanır.
- Basamakların toplamı 15'ten büyük olduğunda, oluşan artan kısım bir sonraki basamağa eklenir.
- **Örnek:**

	A	A	1	2	3	4	F	0
+	1	2	3	4	A	B	C	D
<hr/>								
	B	C	4	6	E	0	B	D

Onaltılık Tabanda Çıkarma İşlemi

- Onaltılık tabanda çıkarma işleminde ikili tabandaki çıkarma işlemleri ile aynı mantık kullanılarak yapılır.
- İkili tabandan farklı olarak 2'ye tümleyen yerine 16'ya tümleyen ve 1'e tümleyen yerine 15'e tümleyen kullanılır.

16'ya Tümleyen Yöntemi ile Çıkarma İşlemi



16'ya Tümlleyen Yöntemiyle Çıkarma İşlemi

- **Örnek:** AA1234F0 - 1234ABCD işlemini 16'ya tümleyen yöntemi kullanılarak aşağıdaki biçimde yapılır:
 - 1234ABCD sayısının 16'ya tümleyenini alarak EDCB5433 değerini elde ederiz.

	A	A	1	2	3	4	F	0
+	E	D	C	B	5	4	3	3
<hr/>								
	1	9	7	D	D	8	9	2 3

- Toplama sonucu, işlenenlerden büyük olduğu için en soldaki bit atılır. Çıkarma sonucu sıfırdan büyüktür.
- Sonuç : $(97DD8923)_{16}$

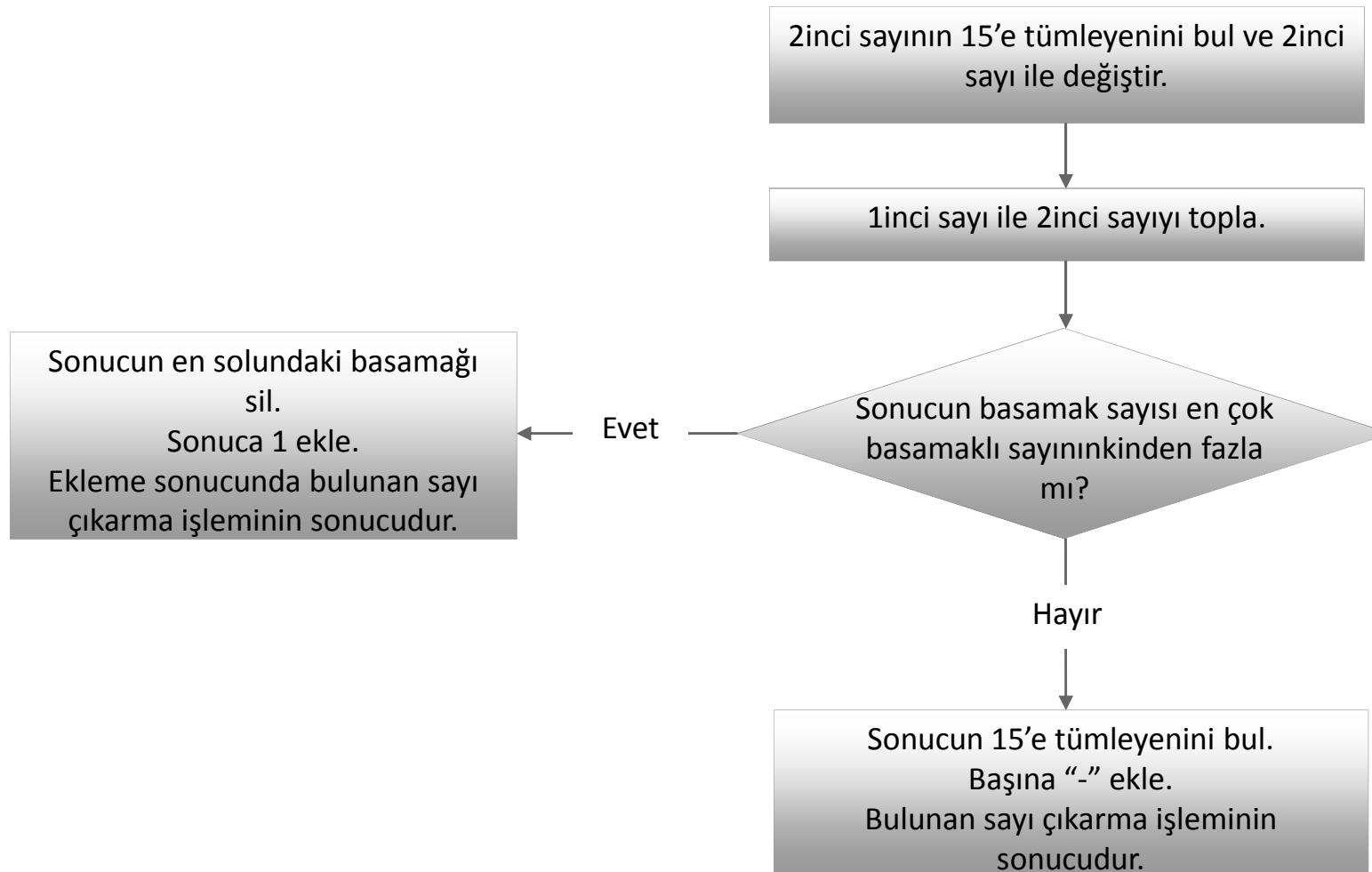
16'ya Tümleyen Yöntemiyle Çıkarma İşlemi

- Örnek: 1234ABCD - AA1234F0 işlemini 16'ya tümleyen yöntemi kullanılarak aşağıdaki biçimde yapılır:
 - AA1234F0 sayısının 16'ya tümleyenini alarak 55EDCB10 değerini elde ederiz.

	1	2	3	4	A	B	C	D
+	5	5	E	D	C	B	1	0
<hr/>								
	6	8	2	2	7	6	D	D

- Toplama sonucu, işlenenlerle aynı sayıda basamağa sahip olduğu için sonucun 16'ya tümleyeni alınır, başına - işareti konur.
- Sonuç : - (97DD8923)₁₆

15'ya Tümleyen Yöntemi ile Çıkarma İşlemi



15'e Tümlleyen Yöntemiyle Çıkarma İşlemi

- **Örnek:** AA1234F0 - 1234ABCD işlemini 15'e tümleyen yöntemi kullanılarak aşağıdaki biçimde yapılır:
 - 1234ABCD sayısının 15'e tümleyenini alarak EDCB5432 değerini elde ederiz.

	A	A	1	2	3	4	F	0
+	E	D	C	B	5	4	3	2
<hr/>								
	1	9	7	D	D	8	9	2

- Toplama sonucu, işlenenlerden büyük olduğu için en soldaki bit atılır. Çıkarma sonucuna 1 eklenir.
- Sonuç : $(97DD8923)_{16}$

15'e Tümleyen Yöntemiyle Çıkarma İşlemi

- **Örnek:** 1234ABCD - AA1234F0 işlemini 15'e tümleyen yöntemi kullanılarak aşağıdaki biçimde yapılır:
 - AA1234F0 sayısının 15'e tümleyenini alarak 55EDCB0F değerini elde ederiz.

	1	2	3	4	A	B	C	D
+	5	5	E	D	C	B	0	F
<hr/>								
	6	8	2	2	7	6	D	C

- Toplama sonucu, işlenenlerle aynı sayıda basamağa sahip olduğu için sonucun 15'e tümleyeni alınır, başına - işareti konur.
- **Sonuç : - (97DD8923)₁₆**

İşaretli Sayılar

- Bilgisayar donanımları yalnızca 0 ve 1 değerleriyle çalışabildiği için, sayının sıfırdan küçük olduğunu gösteren “-” işaretlerinin donanımın anlayacağı biçime sokulmaları gerekir.
- Bu nedenle işaretli sayı gösterimleri tanımlanmıştır.
- Bilgisayarlarda gösterilen ikilik düzendeki işaretli sayılarda **en soldaki bit** sayının işaretini tutar.
 - işaret biti 1 olan sayı sıfırdan küçük
 - işaret biti 0 olan sayı sıfırdan büyük

İşaretli Sayılar

İkilik Tabanda Sayı	İşaret biti sayıdan ayrırsa değeri	1'e tümleyen düzeninde değeri	2'ye tümleyen düzeninde değeri
0000	+0	+0	0
0001	1	1	1
0010	2	2	2
0011	3	3	3
0100	4	4	4
0101	5	5	5
0110	6	6	6
0111	7	7	7
1000	-0	-7	-8
1001	-1	-6	-7
1010	-2	-5	-6
1011	-3	-4	-5
1100	-4	-3	-4
1101	-5	-2	-3
1110	-6	-1	-2
1111	-7	-0	-1

Kayan Nokta (Floatin Point) Veri Türü

- 16 bitlik bir alanda, en soldaki bit işaret amacıyla kullanılırsa, - 2^{15} (-32768) ile $2^{15}-1$ (32767) arasındaki sayılar gösterilebilir.
- Bilgisayarlarda bazı durumlarda, ondalıklı sayılarla ya da çok büyük sayılarla işlem yapmak gerekebilir.
- Böyle durumlarda verilerin gösterilmesi için kayan nokta veri türü kullanılır.
- Kayan nokta veri türünün, çeşitli standartları vardır.
- Günümüz bilgisayarlarında genellikle 32 bitlik bir çeşidi kullanılmaktadır.

IEEE 754 Kayan Veri Türü

- IEEE 754 standardı, 32 bitlik bir veri türünü tanımlamaktadır.
- Bu veri türünde
 - 32 bitin ilk biti işaret,
 - ilk bitten sonraki sekiz bit üs (exponent)
 - bu dokuz bitten sonra gelen yirmi üç bit anlamlı kısım (mantissa, fraction)
- Gösterilmek istenen sayı $1,aaa.. \times 2^{\ddot{u}}$ biçimine getirildiğinde,
 - aaa.. değeri anlamlı kesimi
 - \ddot{u} değeri üs kesimini oluşturur.

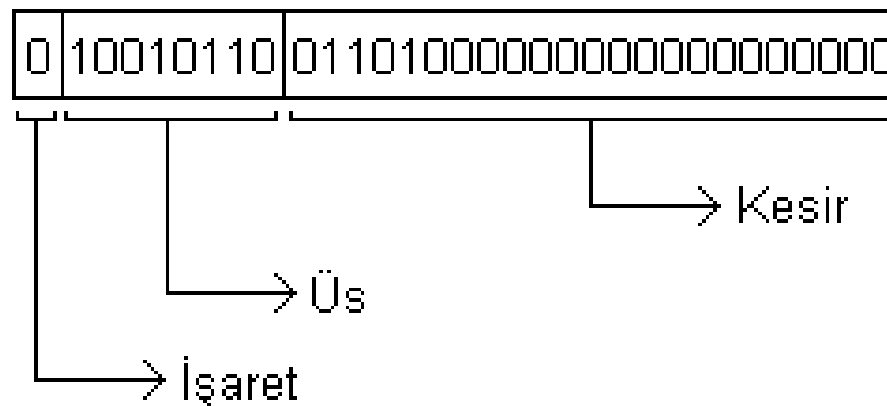
IEEE 754 Kayan Veri Türü

İşaret (1 bit) = A	Üs (8 bit) = B	Anlamalı Kısım (23 bit) = C
--------------------	----------------	-----------------------------

- IEEE 754 biçiminde gösterilen yukarıdaki 32 bitlik bir verinin değeri şu şekilde hesaplanabilir:

$$\text{Sayının Değeri} = (-1)^A \times 1, C \times 2^{B - 127}$$

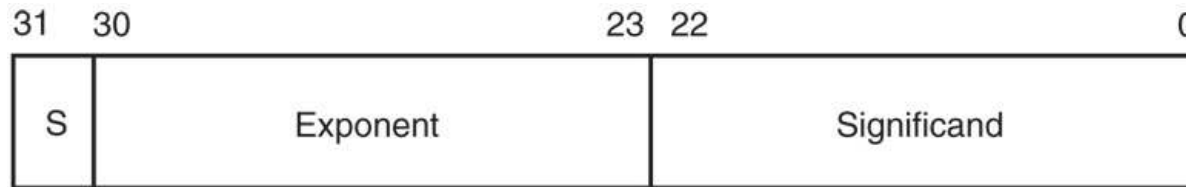
- Örnek:** $1,01101 \times 2^{23}$ sayısı IEEE 754 kayan nokta veri türü biçiminde 32 bitle aşağıdaki gibi gösterilir:



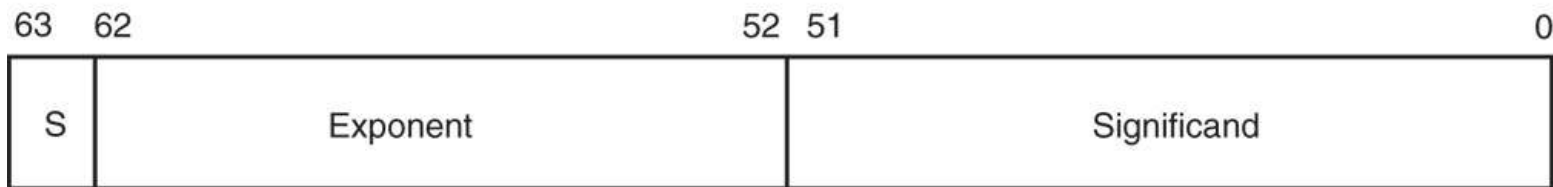
IEEE 754 Kayan Veri Türü

- **Örnek:** IEEE 754 kayan nokta veri türü biçiminde gösterilen 10101010101111110111000000000000 sayısının değeri aşağıdaki gibi bulunur.
 - En soldaki (31.) bitin değeri 1 olduğu için, sayının sıfırdan küçüktür.
 - Ardından gelen 8 bitin değeri 01010101 olduğundan $B = 85$ alınır.
 - Buradan $B - 127 = -42$ bulunur.
 - Üs bitlerinden sonraki 011111101110000000000000 sayısı ,virgülden sonraki kesimdir.
 - Böylece sayının değeri:
$$(-1)^1 \times 1,011111101110000000000000 \times 2^{-42} = -1,01111110111 \times 2^{-42}$$

Tek ve Çift Duyarlıklı Kayan Noktalı Sayılar (Single and Double Precision Floating Point Numbers)



(a)

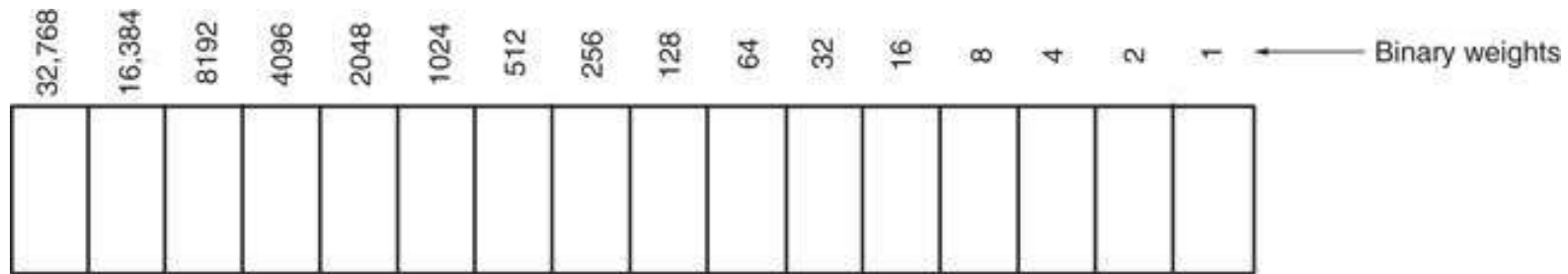


(b)

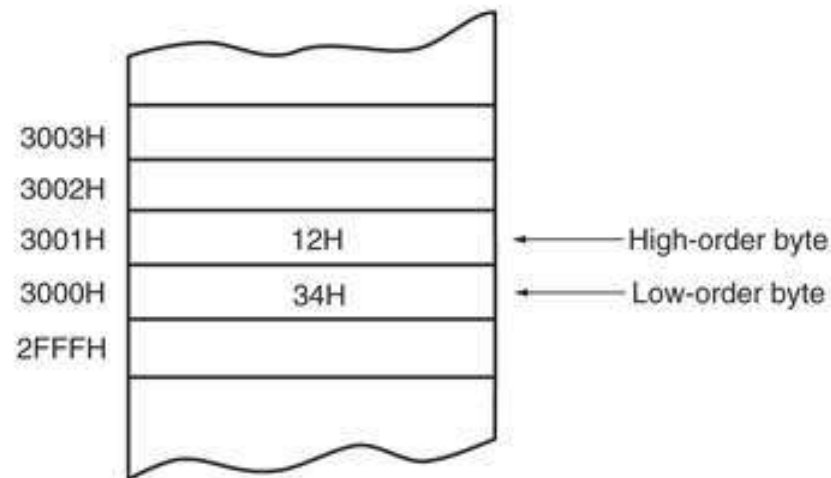
Düşük Öncelikli Bit Önce (Little Endian) Gösterimi

- Birden fazla byte'dan oluşan sayıların ana bellekte tutulmasında, düşük öncelikli bitler ana bellekte önce yerleşirse, bu gösterime little endian gösterimi denir.

Düşük Öncelikli Bit Önce (Little Endian) Gösterimi

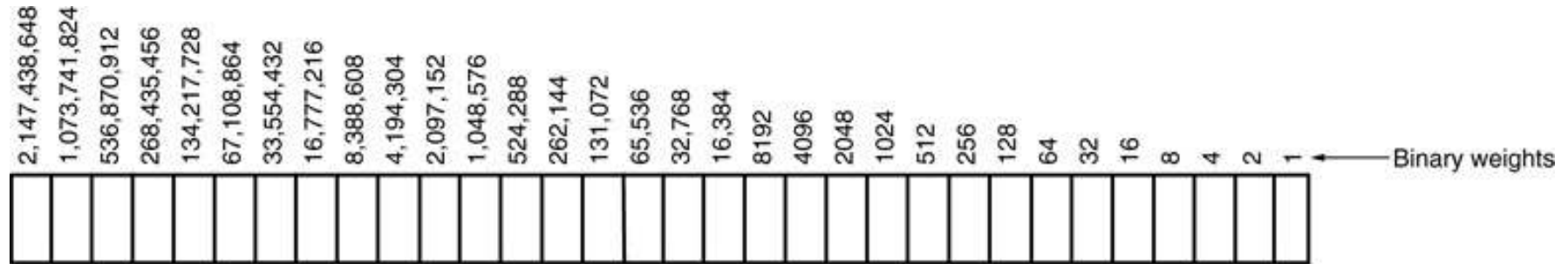


(a) Unsigned word

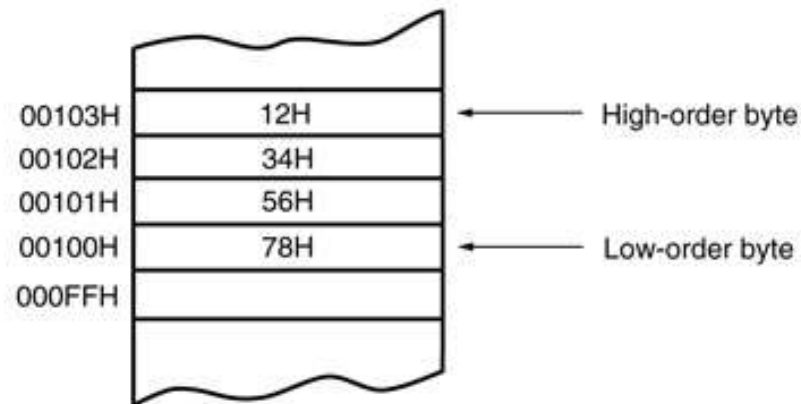


(b) The contents of memory location 3000H and 3001H are the word 1234H.

Düşük Öncelikli Bit Önce (Little Endian) Gösterimi



(a) Unsigned doubleword



(b) The contents of memory location 00100H–00103H are the doubleword 12345678H.

Yüksek Öncelikli Bit Önce (Big Endian) Gösterimi

- Birden fazla byte'dan oluşan sayıların yüksek öncelikli bitleri ana belleğe önce yerleştirilirse, bu gösterime big endian gösterimi denir.
- Little endian gösteriminin tam tersi gösterimi ifade eder.