2-Sayı Sistemleri

Sayı Sistemleri

- Herhangi bir tabanda ki sayı, o tabanda kullanılabilecek rakamların belirli kurallar ile yan yana getirilmesinden oluşmuştur. Bu oluşum D.1'de verilmiştir. Bu denklem aynı zamanda, herhangi bir sayı tabanında yazılmış bir sayının 10 tabanındaki karşılığını da elde etmek için kullanılır.
- Burada T kullanılan sayı tabanı, i ve j basamak değeri, A ise sayı tabanında kullanılabilecek rakamlardır. A, 0 ile T-1 arasındaki herhangi bir rakam olabilir.

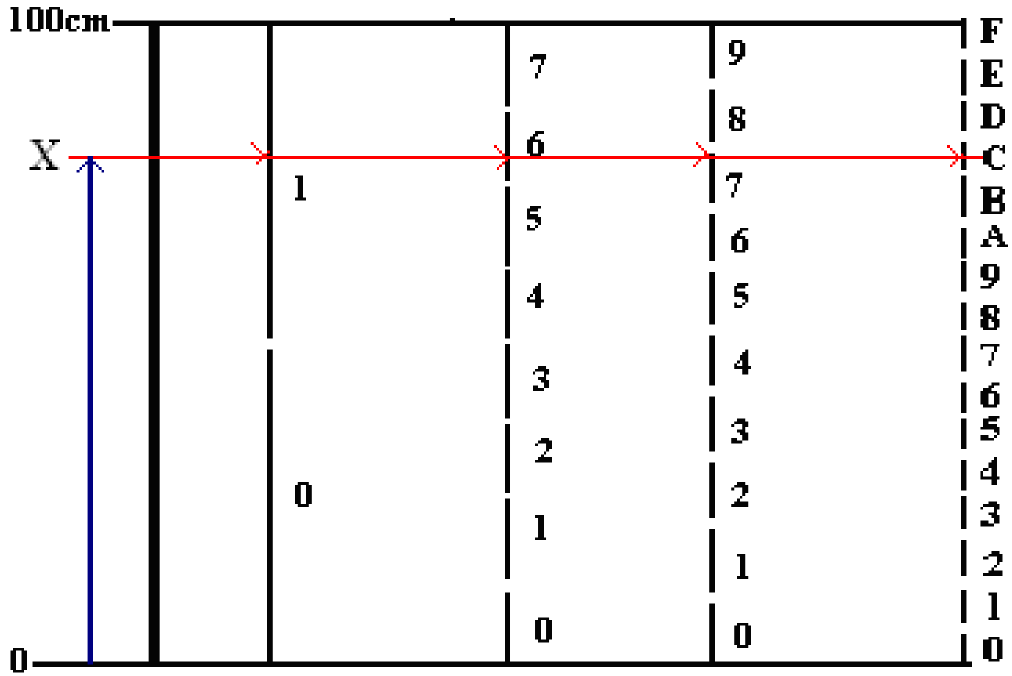
$$\mathbf{(546{,}13)10 = 5x10^{2} + 4x10^{1} + 6x10^{0} + 1x10^{-1} + 3x10^{-2}}$$
$$\mathbf{= 500 + 40 + 6 + .10 + 0.03 = 546{,}13}$$

$$(\text{Sayı})_T = \sum_{i=0}^{i=n-1} A_i \cdot T^i + \sum_{j=-m}^{j=-1} A_j \cdot T^j \qquad \text{D.1}$$

(Tamsayı kısmı) + (kesirli kısmı)

- Günlük hayatta kullandığımız ondalık (On tabanlı - Desimal) sayı sisteminin yanı sıra, sayısal sistemler için 2 tabanlı, 8 tabanlı, 16 tabanlı, 32 tabanlı sayı sistemleri de çok önemlidir.
- Değişik sayı tabanlarında tek basamakla ifade edeceğimiz sayılar veya rakamlar 0,….T-1 arasında herhangi biri olabilir. Diğer bir değişle tek basamakla ifade edebileceğimiz farklı büyüklük sayısı taban değeri kadardır.
- Örneğin tek basamaklı olarak, 2 tabanlı sayı sisteminde 2, 8 tabanlı sayı sisteminde 8, 10 tabanlı sayı siteminde 10, 16 tabanlı sayı sisteminde 16 farklı sayı yazılabilir. Veya farklı büyüklük ifade edilebilir

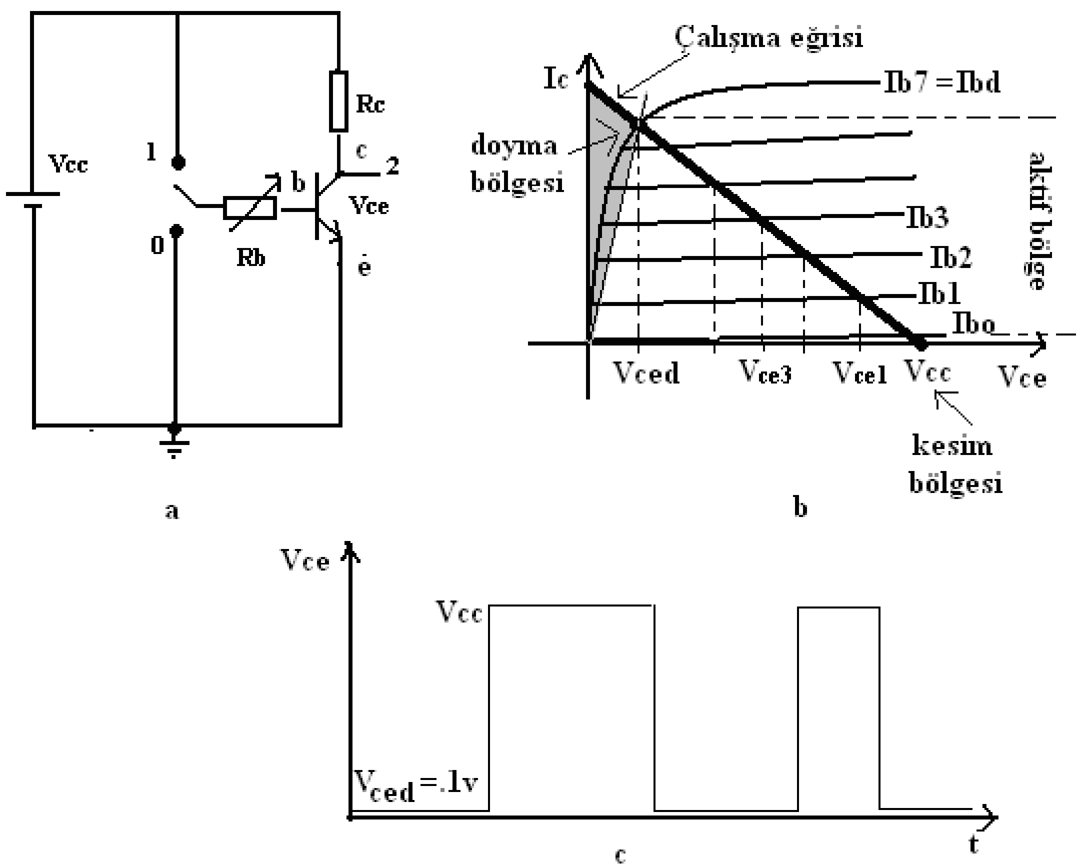
X noktasının 0 noktasına göre uzaklığının değişik sayı sistemlerinde tek basamaklı sayılarla ifade edilmesi görülmektedir (78,125 cm)



	Bölme sayısı	Bölme Hassasiyeti	X ifadesi	X ölçümü
2 tabanlı s.s	2	100*1/2 = % 50	1	51 <X >100 cm
8 tabanlı s.s	8	100*1/8 = % 12.5	6	75 <X > 87.5 cm
10 tabanlı s.s	10	100*1/10 = % 10	7	70 <X > 80 cm
16 tabanlı s.s	16	100*1/16 = % 6.5	C	78 <X >84.5 cm

2.1. Sayısal Sistemlerde Sayı Tabanının Önemi

- Herhangi bir büyüklüğün ifadesinde kullanılan sayı sisteminin tabanı ne kadar büyük olursa, büyüklüğün doğru ifade edilme oranı o derece yükselir. Hassasiyet artar
- Ancak sayısal sistem elektroniğinin rakamları tanıması problemi ortaya çıkar.



- **Eğer sayısal sistem 2 tabanlı sayı sistemine göre düzenlenirse; bu durumda sadece 0 ve 1 rakamlarının tanınması için zaman üzerinde sadece iki değer alabilen bir gerilim işareti yeterli gelir. Bu, transistorün anahtarlama modunda çalışmasıyla elde edilir.**
- **Sayısal sistemde Octal sayı sistemi kullanılıyorsa, 8 farklı rakamın sisteme tanıtılması gerekir. Bunun anlamı, zaman üzerinde 8 farklı gerilim seviyesi bulunan bir işaret üretmektir. VeyabTransitörün aktif bölgede 8 farklı çalışma noktasında çalışması anlamındadır. Bu çalışma şartı ZORDUR**
- **Hexadecimal sayı sistemi ile doğrudan çalışılacaksa 16 farklı rakam için 16 farklı gerilim seviyesi elde edilmelidir. Bu durumun gerçekleştirilmesi daha da zordur**
- **Bu önemli gerçekleştirme problemlerinden ötürü, sayısal sistemlerin temelinde 2 tabanlı sayı sitemi kullanılmaktadır. Çünkü bu tabanda kullanılan 1 ve 0 rakamlarını donanımsal olarak tanımak için, zamana göre iki durumda olan işaret üretmek, transistorün anahtarlama modunda çalıştırılmasıdır. Çok kolaydır.**
- **Ancak, yukarıda belirtildiği gibi sayı tabanının büyük olmasının birtakım avantajları vardır. Sayısal sistemlerde bu avantajlardan yararlanmak için; büyük tabanlı sayı sistemlerindeki rakamların 2 tabanlı sayı sisteminde kodlanması ve kullanılması önemli bir çözümdür ve kullanılmaktadır.**
- **2 tabanında n bit ile 2^n tane değişik sözcük oluşturulabilir. *Kullanılabilecek rakam sayısı, 2'lik sistemdeki 2^n tane farklı değişik sözcük sayısı kadar olan sayı tabanları tam kodlamalı sayı sistemi olarak adlandırılmaktadır.* Örneğin 8 tabanlı sayı sistemindeki rakamlar n=3 bit ile, Hexadecimal sayı sisteminde rakamlar n=4 bit ile artıksız olarak ifade edilir.**
- **Tam kodlamalı sayı sistemlerinin başka avanyajları da yeri geldikçe açıklanacaktır.**
- **Oysa on tabanlı sayı sitemindeki 10 farklı rakamı 2'lik sistemde kodlamak için 4 bit'e ihtiyaç vardır. Fakat bu 4 bitlik sözcüklerden (2^4 = 16 adet) sadece on tanesi kullanıldığı için desimal sayı sistemi tam kodlamalı sayı sistemi değildir.**

2.3. Sayısal Sistemler için önemli olan Sayı Tabanları

- On tabanlı (Desimal) sayı sistemi günlük hayatta kullandığımız fakat sayısal sistemler için uygun olmayan bir sayı sistemidir. Bunun yanında 2 tabanlı sayı sistemi ve tam kodlamalı 8 tabanlı (Octal) ve 16 tabanlı (Heksadesimal) sayı sistemleri bizim için oldukça önemlidir. Bu bölümde bu sayı sistemlerinde sayıların oluşumu ve kendi aralarındaki dönüşümler açıklanacaktır.

- **2.3.1. On tabanlı (Decimal) Sayı Sistemi**
- Günlük hayatta kullanılan sayı sistemdir. Katsayı rakamları 0, (T-1)'e göre 0,1,2,3,4,5,6,7,8,9 gibi on farklı rakamdır. Sayı oluşumu D.1'e göre yapılır. Tam kodlamalı bir sayı sistemi olmadığından sayısal sistemler için uygun değildir. Bu tabanda yazılmış **S = (547,75)10** sayısının oluşumu aşağıdaki gibidir.

- $\mathbf{S = 5x10^{2} + 4x10^{1} + 7x10^{0} + 7x10^{-1} + 5x10^{-2} = 500 + 40 + 7 + .7 + .05}$
  $\mathbf{= 547,75}$

İki Tabanlı (Binary) Sayı Sistemi

- Bu sayı sisteminin tabanı 2'dir ve kullanılan rakamlar sadece 0 ve 1'dir. Bu rakamların yan yana getirilmesiyle sayı oluşturulur. Binary sayının her bir basamağına **Bi**nary Digi**t**'in kısaltılması olan **Bit** denir. Sayı oluşturmak için D.1'deki kural geçerlidir.
- Örneğin **S = (11001.11)2** sayısının oluşumu aşağıdaki gibidir.
- **S = $1x2^4 + 1x2^3 + 0x2^2 + 0x2^1 + 1x2^0 + 1x2^{-1} + 1x2^{-2}$ = 16+8+ 0 + 0 +1+ .5 + .25 = (25.75)10**

- İki tabanında yazılmış sayının en soldaki Bit’ine En önemli Bit (Most Significient Bit) **MSB** ve en sağdaki Bit’ine ise En Önemsiz Bit (Least Significient Bit) **LSB** bit’i denir. Bununla ilgili örnek aşağıda görülmektedir.
- **1 0 0 0 1 1 1**

MSBbiti **LSB biti**

Sekiz Tabanlı (Octal) Sayı Sistemi

- 8 tabanlı sayı sisteminde kullanılabilen rakamlar 0,1,2,3,4,5,6,7 ‘dir.
- Sayı oluşumu D.1’e göredir. Tam kodlamalı bir sistem olduğu için önemlidir. Örnek olarak;
- S = $(176)_8$ sayısının oluşumu aşağıdaki gibidir.
- **S = $1x8^2 + 7x8^1 + 6x8^0 = 64 + 56 + 6 = (126)_{10}$**
- İlk oluşturulan sayısal sitemlerde ve bilgisayarlarda kullanılmasına karşılık günümüz sistemlerinde terkedilmiştir.

Onaltı Tabanlı (Hexadecimal) Sayı Sistemi

- 16 tabanlı sayı sisteminde kullanılabilen rakamlar; **0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F** şeklindedir.
- Tam kodlamalı bir sayı sistemi olduğu için önemlidir. Bu tabandaki **$S=(1AC7)_{16}$** sayısının oluşumu D.1 bağıntısına göre aşağıda verilmiştir.
- **$S = 1x16^3 + 10x16^2 +12x16^1 + 7x16^0 = 4096 + 2560 + 07 = (6663)_{10}$**

Sayı tabanları arasındaki dönüşümler

- Herhangi bir tabandaki sayının, günlük hayatta kullanılan 10 tabanlı sayı sistemine dönüşümü D.1 eşitliği ile kolay bir şekilde elde edilir.
- Genel olarak ***n tabanlı*** sayı sisteminde yazılmış bir sayının ***T tabanına*** dönüşümü için yapılması gereken; n tabanlı sayının devamlı olarak T taban değerine bölünerek bu işlemin, ***bölüm***ün T'den küçük oluncaya kadar devam ettirilmesidir.
- Bu bölme işlemi n tabanlı sayı sistemi kurallarına göre olmalıdır. Her bölme işleminde ***kalan*** değer, yeni sayı sistemindeki sayının rakamlarını oluşturur.
- İlk bölmedeki ***kalan*** değer yeni sayıdaki en ağırlıksız basamaktır. En son ***bölüm*** değeri yeni sayının en soldaki basamağını oluşturur

	bölüm	kalan
$\frac{133}{2}$	$= 66$	1
$\frac{66}{2}$	$= 33$	0
$\frac{33}{2}$	$= 16$	1
$\frac{16}{2}$	$= 8$	0
$\frac{8}{2}$	$= 4$	0
$\frac{8}{2}$	$= 2$	0
$\frac{2}{2}$	$= 1$	0

10000101

Decimal→Binary

$$(133)_{10} = (10000101)_2$$

	bölüm	kalan
$\frac{133}{8}$	$= 16$	5
$\frac{16}{8}$	$= 2$	0

2 0 5

Decimal→octal

$$(133)_{10} = (205)_8$$

	bölüm	kalan
$\frac{133}{16}$	$= 8$	5

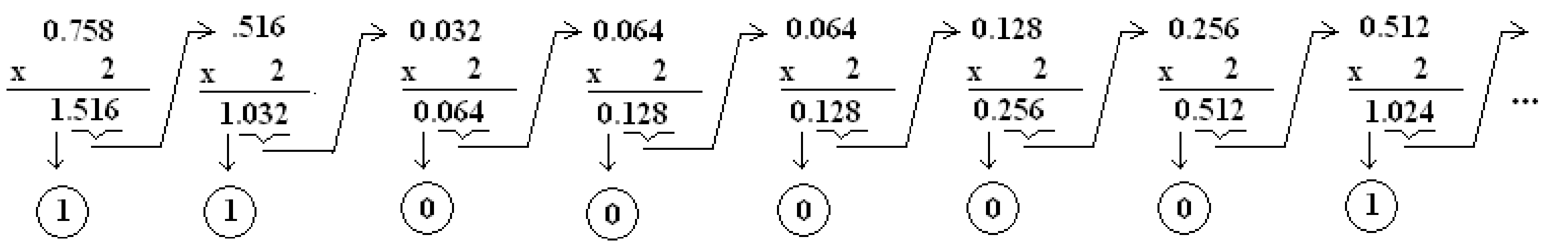
8 5

Decimal→hexadecimal

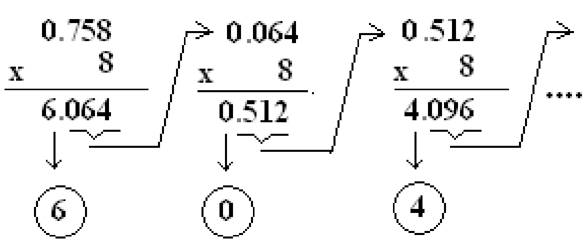
$$(133)_{10} = (85)_{16}$$

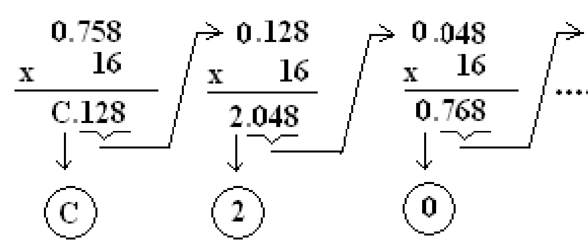
Kesirli sayılarda dönüşüm

- **Bu durumda n tabanında yazılmış kesirli sayı, T tabanında ifade edilecekse, sayı devamlı olarak T tabanıyla çarpılır, çarpımın tam sayı kısmı alınarak T tabanındaki kesirli sayının rakamları oluşturulur. İlk çarpımdan elde edilen rakam T tabanındaki kesirli sayının en soldaki basamağını teşkil eder. Çarpma adedi, elde edilen kesirli sayının hassasiyetiyle orantılıdır. 0.758 sayısının değişik tabanlara dönüşümü aşağıdadır.**
- 10 tabanındaki ***133,758*** sayısının, İki tabanlı karşılığının ***10000101.1100001***, 8 tabanındaki karşılığının ***205.604*** ve 16 tabanındaki karşılığının ise ***85.C2*** olduğu kolayca görülebilir.



$(0.758)_{10} = (.1100001)_2$





$(0.758)_{10} = (.604)_8$

$(0.758)_{10} = (.C20)_{16}$

- 10 tabanlı sayıların, diğer tabanlarda ifade edilmesi yukarıda verilen sistematik biçimde gerçekleştirilebilir.
- Ancak diğer tabanlar arası dönüşümlerde de yukarıdaki sistematik kural, yani devamlı T tabanına bölme kuralı geçerlidir.
- Fakat değişik sayı sistemlerindeki bölme işlemi kuralları (Çarpma ve çıkrama) alışkanlığımız dışında olduğundan zor gelebilir. Bu durumu dört işlem matematiği konusunda daha detaylı göreceğiz.

Sayı tabanları arasında dönüşüm için başka bir yol

- Sayıyı 10 tabanına çevirip sonra işlem yapmak olabilir. Örneğin $(FF)_{16}$ sayısının 8 tabanındaki karşılığını bulmak için; sayı önce 10 tabanına çevrilerek;

  $$\mathbf{(FF)_{16} = Fx16^1 + Fx16^0 = 15x16 + 15x1 = (255)_{10}}$$

  elde edilir. Sonra 255 sayısının 8 tabanındaki karşılığı için devamlı 8'e bölünüp, bölümün 8'den küçük oluncaya kadar, işleme devam edilip düzenlenirse;

- $(FF)_{16} = (255)_{10} = (377)_8$ elde edilir.

- Aynı şekilde $(FF)_{16}$ sayısının 2 tabanındaki karşılığı için $(255)_{10}$ sayısına, T'a bölme kuralı uygulanarak

  $(FF)_{16} = (255)_{10} = (11111111)_2$ elde edilir.

Pratik dönüşümler

- **Taban dönüşümleri çoğunlukla tam kodlamalı sayı sistemleri ile Binary sistemleri arasında veya tersi olarak yapılır. Tam kodlamalı 2, 8, 16 tabanlı sistemler arasındaki dönüşümlerin pratik olarak kolaylıkla nasıl yapılacağını görelim.**

- **Binary ↔ Octal dönüşümü : Octal taban tam kodlamalı sayı sistemi olduğundan, sayıları oluşturan 8 tane farklı rakamın her biri 3 bitlik 8 binary sözcükle ifade edilir. Bu durum Tablo 2.2'de görülmektedir. Octal sistemde yazılmış bir sayının ikilik sisteme dönüştürülmesi için bu sayının rakamların 3 bitlik karşılıklarının yan yana yazılması yeterlidir. Örnek:**

- **$(5437023)_8$ = (101 100 011 111 000 010 011)2**

Tablo 2.2. 8 tabanlı sistemdeki rakamların Binary sözcüklerle ifadesi

0	1	2	3	4	5	6	7
000	001	010	011	100	101	110	111

Pratik dönüşümler-2

- Binary sayıdan, Octal sisteme dönüşüm için; sözcük sağdan itibaren 3'er bit olarak guruplanır. Her 3 bitin octal sistemdeki rakam karşılıklarının yan yana yazılmasıyla Sayının Octal sistemdeki karşılığı elde edilir.

$$\underbrace{1}_{1}\;\underbrace{101}_{5}\;\underbrace{100}_{4}\;\underbrace{111}_{7}\;\underbrace{010}_{2}\;\underbrace{110}_{6}\;\underbrace{110}_{6}\;\underbrace{111}_{7}\;\underbrace{001}_{1}\;\underbrace{101}_{5}\;\underbrace{011}_{3} = (15472667153)_8$$

$$5437023$$

$$\overline{101}\;\overline{100}\;\overline{011}\;\overline{111}\;\overline{000}\;\overline{010}\;\overline{011}$$

Binary-hexa dönüşümü

- **Binary ↔ Hexadecimal dönüşümü:** Hexadesimal sistem tam kodlamalı sistem olduğundan, sayıları oluşturan 16 farklı rakam, binary sitemde 4 bitlik 16 farklı sözcüğe karşılık düşürülür. Bu durum tablo 2.3'de görülmektedir. Hexadecimal sistemde yazılmış bir sayının ikilik sisteme dönüştürülmesi için bu sayının rakamlarının 4 bitlik binary karşılıklarının yan yana yazılması yeterlidir. Örnek

$$(FB90C37)_{16} = (1111\ 1011\ 1001\ 0000\ 1100\ 0011\ 0111\ )_2$$

Tablo 2.3. **16 tabanlı sistemdeki rakamların Binary sözcüklerle ifadesi**

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

Binary-Hexa dönüşümler -2

- Binary sistemde yazılmış herhangi bir sayının 16 tabanlı sayıya dönüştürülmesi için ise; sözcük sağdan itibaren 4'er bit olarak ayrılır. Bunlar 16'lık sistemin her bir rakamının binary kodlanmış şeklidir. Yan yana yazılarak 16 Tabanındaki sayı elde edilir.

$$\underbrace{110}_{6}\,\underbrace{1100}_{C}\,\underbrace{1110}_{E}\,\underbrace{1011}_{B}\,\underbrace{0110}_{6}\,\underbrace{1110}_{E}\,\underbrace{0110}_{6}\,\underbrace{1011}_{B} = (6CEB6E6B)_{16}$$

FB90C37

1111 1011 1001 0000 1100 0011 0111

Tablo 2.3. 10,8,16 Tabanlı bazı sayıların 2'lik tabanda ifadesi

Decimal		Octal		Hexadecimal	
0	0001	0	000	0	0001
1	0001	1	001	1	0001
2	0010	2	010	2	0010
3	0011	3	011	3	0011
4	0100	4	100	4	0100
5	0101	5	101	5	0101
6	0110	6	110	6	0110
7	0111	7	111	7	0111
8	1000	10	001 000	8	1000
9	1001	11	001 001	9	1001
10	1010	12	001 010	A	1010
11	1011	13	001 011	B	1011
12	1100	14	001 100	C	1100
13	1101	15	001 101	D	1101
14	1110	16	001 110	E	1110
15	1111	17	001 111	F	1111
16	1 0000	20	010 000	10	0001 0000

İşaretli Sayılar

1- İşaret –Mutlak değer gösterimi

- Genellikle iki tabanlı sayı sistemi içindir. Sayının mutlak değeri pozitif ve negatif sayı için aynıdır. Sayı sözcüğünün MSB biti işaret bitidir. 0 ise sayı pozitif, 1 ise sayı negatifdir. Örneğin $(+27)_{10}$ ve $(-27)_{10}$ sayılarını, ikilik sistemde 8 bitlik sözcüklerle ifade edelim.

- (+27)10 → **0** 0011011
- (-27)10 → **1** 0011011

- Bu tip ifade tarzı en kolay ifade tarzıdır. Bu durumda iki tane 0 gösterimi oluşmaktadır. Bunun yanı sıra özellikle çıkarma işlemi için sonuçta önemli düzeltmelerin yapılması gerekmektedir. Bu da sistemin donanımsal yapısını karmaşıklaştırdığı için kullanımından vazgeçilmiştir.

2-) Taban -1’e göre gösterim

- T tabanlı n basamaklı bir N sayısının Taban -1’e göre tümleyeni $(T^n -1) - N$ bağıntısı ile elde edilir.

Örneğin; $(627)_{10}$ sayısının Taban -1’e tümleyeni; $(10^3 -1) - 627 = (1000 - 1) - 627 = 372$ ‘dir. 372 sayısı Taban -1’e tümleme sisteminde

-627’yi ifade etmektedir.

- Taban-1’e göre tümleme pratik olarak her basamaktaki rakamın (T-1)’den çıkarılmasıyla da yapılır.

Örneğin; $(563)_8$ sayısının Taban-1’ göre tümleyeni: $777 - 563 = (214)_7$ olarak veya $(AB3)_{16}$ sayısının T-1’e tümleyeni,

$FFF - AB3 = (54C)_{15}$ olarak elde edilir.

2-) Taban -1'e göre gösterim

- Sayısal sistemler için çok daha önemli olan 2 tabanlı sayı sisteminde, işaretli sayıları ifade etmek için;
- Sayı pozitif ise işaret-mutlak değer gösterimi geçerlidir.
- Sayı negatif ise pozitif sayının işaret-mutlak değer gösteriminin Taban-1'e göre tümleyeni alınır.
- İki tabanlı sayı sisteminde, pratik olarak Taban -1'e göre tümleme için sözcükteki 1'erin yerine 0, 0'ların yerine 1 koymak yeterlidir. İkilik sistemde Taban – 1'e tümlemenin adı 1'e tümleme'dir. Aşağıda bununla ilgili örnekler görülmektedir.
- $(+3)_{10}$ = **0** 0011, $(-3)_{10}$ = **1** 1100
- $(+15)_{10}$ = **0** 1111, $(-15\ )_{10}$= **1** 0000
- İkilik sistemde 1'e tümleme metoduna göre yazılmış sayıların çözümlemesi şöyle olmaktadır;
- Pozitif sayılar mutlak değer kısmının D.1 formülüne göre elde edilmesiyle değerlendirilir.
- Negatif sayılarda ise işaret bitinin basamak değerine negatif değer verilerek, işaretli sayının diğer bitlerinin D.1'e göre bulunan değeri, bu negatif değer ile toplanıp sonuca 1 eklenirse negatif sayının değeri elde edilmiş olur. Aşağıdaki örnekleri inceleyiniz.

- + 14 = **0** 1110 = $1x2^3 + 1x2^2 + 1x2^1 + 0x2^0$ = + 14
- \- 14 = **1** 0001 = -16 +1 = - 15 +1 = - 14

3- Tabana göre tümleme ile gösterim

- En çok kullanılan gösterim tarzıdır.

  $$T^{n} - N$$

  bağıntısı ile negatif sayıların gösterimi yapılır.

  Örneğin, $(-2389)_{10}$ sayısının tabana tümlenmesi;

  (10000 – 2389) = 7611 şeklindedir.
- İkilik sistemdeki işaretli sayılardan, pozitif olanlar işaret-mutlak değer gösterimindeki gibi ifade edilir.
- Negatif sayılar ise, pozitif sayının işaret- mutlak değer gösteriminin Taban'a göre tümlenmesi ile elde edilir. Binary sistemde,Taban'a göre tümleme işlemi 2'ye tümleme olarak adlandırılır. Pratik olarak 2'ye tümlenecek sayı 1'e tümleme metoduna göre yapılıp elde edilen sözcüğe 1 eklenir. Örnek olarak;
- (+ 14)10 = **0** 1110
- (- 14)10 = **1** 0001 + 1 = 1 0010

2’ye tümleme’de dönüşüm

- Tabana tümleme metodundaki ikilik pozitif sayıların onluk değerlerini bulmak için D.1 bağıntısı kullanılır.
- Negatif sayılar ise, işaret bitinin onluk değerini negatif alarak, diğer bitlerin oluşturduğu onluk sayı sistemindeki sayı ile toplanması ile bulunur. Aşağıdaki örneği inceleyiniz.

- $(+14)_{10} = 0\,1110 = 0x2^4 + 1x2^3 + 1x2^2 + 1x2^1 + 1x2^0 = 8 + 4 + 2 = +14$
- $(-14)_{10} = 1\,0010 = -(1x2^4) + (0x2^3 + 0x2^2 + 1x2^1 + 1x2^0) = -16 + 2 = -14$

Yorum

- İşaret-mutlak değer gösterimi çok kolay olmasına karşılık, her sayı için mutlak değer ile birlikte işaret bitinin de kontrol edilmesi gerekir. Ayrıca iki tane sıfır tanımı vardır. Yine bu tür gösterimde aritmetik işlemlerde birtakım zorluklar çıkmaktadır.
- Taban-1'e tümleme metodunda da iki tane sıfır tanımı mevcuttur. Negatif sayıları elde etmek için sonuca 1 eklenmesi gerekmektedir. Aritmetik işlemlerin sonucunda da 1 eklenmesi gerekebilir.
- Tabana tümleme metodunda ise tekbir 0 tanımı ve aritmetik işlemlerde ilk iki yönteme göre kolaylıklar mevcuttur. Bu yüzden sayısal sistemlerde işaretli sayılar için tabana tümleme yöntemi kullanılmaktadır. 2'ye tümleme sisteminde *n*-bitle gösterilebilecek sayılar,

$$-(2^{n-1}) \text{ ile } +(2^{n-1})-1$$

aralığındadır.

Buna karşılık *n*-bit'le gösterilebilecek işaretsiz sayı 2^n tanedir.

Örnekler:

- n=3 için→ $-(2^{3-1})$ ile $+(2^{3-1})-1)$ = - 4 ile +3 aralığında.
- n=4 için→ $-(2^{4-1})$ ile $+(2^{4-1})-1)$ = - 8 ile +7 aralığında.
- n=6 için→ $-(2^{6-1})$ ile $+(2^{6-1})-1)$ = - 32 ile +31 aralığında sayılar elde edilir.

Tablo2.4. 3 bitlik sözcüklerin işaretsiz ve işaretli sayı olarak değerlendirilmesi

n=3	İşaretsiz sayı	İşaret-Mutlak değer	Taban -1'e tümleme	Taban'a tümleme
000	0	+0	+0	+0
001	1	+1	+1	+1
010	2	+2	+2	+2
011	3	+3	+3	+3
100	4	-0	-3	-4
101	5	-1	-2	-3
110	6	-2	-1	-2
111	7	-3	-0	-1

Sayı Sistemlerinde Aritmetik İşlemler

- **<u>İşaretsiz sayılarla Toplama İşlemi</u>:** *Toplama işlemine, sayıların en sağdaki basamaklarından başlanır. Basamaktaki toplama sonucu tabanı aşmışsa bir üst basamağa elde(taban değeri) olarak sunulur. Kalan; işlem basamağındaki toplama sonucudur.*
- ***İki tabanlı Sayı Sisteminde toplama***

  İşaretsiz sayılarla toplamadaki kural (herhangi bir basamaktaki iki bitin toplanması);

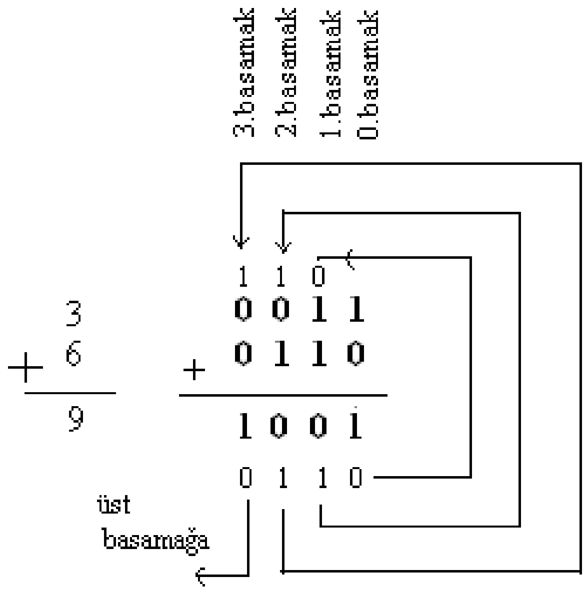
  0 + 0 = 0

  0 + 1 = 1

  1 + 0 = 1

  1 + 1 = 0 elde 1 **(oluşan elde biti bir üst basmağa verilir).**

Toplama Örnekleri



0.basamak	1+0=1	Elde 0
1.basamak	1+1=0	Elde 1
2.basamak	0+1+1=0	Elde 1
3.basamak	0+0=0	Elde 0

```
  12        1100
  15        1111
+ 07      + 0111
  34       10010
```

```
  126       1111110
+ 127     + 1111111
  253      11111101
```

```
  9.5       1001.10
+ 3.5     + 0011.10
 13.0       1101.00
```

```
  96
+ 23
```

```
  345
+ 912
```

```
  8073
+ 1024
```

Sekiz ve Onaltı tabanlı sayı sistemlerinde toplama

- Sayısal sistemler için önemli olan 8 ve 16 tabanlı ve günlük kullandığımız 10 tabanlı sayı sistemlerindeki toplama işlemi için de yukarıdaki genel kural geçerlidir.Yani İşlem basamağındaki sonuç, tabanı aşmışsa taban değeri üst basamağa verilir, kalan değer o basamaktaki sonuçtur. Bununla ilgili örnekler aşağıdadır

$$\begin{array}{r} 27_8 \\ +\ 16_8 \\ \hline 45 \end{array} \qquad \begin{array}{r} 27_{10} \\ +\ 16_{10} \\ \hline 43_{10} \end{array} \qquad \begin{array}{r} 27_{16} \\ +\ 16_{16} \\ \hline 3D_{16} \end{array} \qquad \begin{array}{r} 67_8 \\ +\ 32_8 \\ \hline 121_8 \end{array} \qquad \begin{array}{r} 67_{10} \\ +\ 32_{10} \\ \hline 99_{10} \end{array} \qquad \begin{array}{r} 67_{16} \\ +\ 32_{16} \\ \hline 99_{16} \end{array}$$

$$\begin{array}{r} 326_8 \\ +\ 657_8 \\ \hline 1205_8 \end{array} \qquad \begin{array}{r} 326_{10} \\ +\ 657_{10} \\ \hline 983_{10} \end{array} \qquad \begin{array}{r} 326_{16} \\ +\ 657_{16} \\ \hline 97D_{16} \end{array} \qquad \begin{array}{r} 70_8 \\ +\ 53_8 \\ \hline 143_8 \end{array} \qquad \begin{array}{r} 56_{10} \\ +\ 43_{10} \\ \hline 99_{10} \end{array} \qquad \begin{array}{r} 38_{16} \\ +\ 2B_{16} \\ \hline 63_{16} \end{array}$$

Octal ve Hexadecimal sayıların Binary sistemde toplanması

- Sayısal sistemlerinde 2 tabanlı sayı sistemi kullanılır. O halde yukarıdaki değişik tabanlı sayı sistemlerinde yazılmış sayıları, sadece 2 tabanlı toplama yapabilen bir donanımla toplamak için; bu sayıları ikilik olarak ifade etmek, iki tabanında toplamak ve sonucu istenilen sayı tabanlarına dönüştürmek yeterlidir. Bu işlemler için kodlayıcı, kod açıcı ve ikilik tabanda toplama yapabilen donanımsal ünite gerekir.

$$\begin{array}{r} 27_8 \\ +\ 16_8 \\ \hline 45_8 \end{array} \qquad \begin{array}{r} 010\ 111 \\ +\ 001\ 110 \\ \hline \underline{100}\ \underline{101} \\ 4\quad\ 5 \end{array}$$

$$\begin{array}{r} 27_{10} \\ +\ 16_{10} \\ \hline 43_{10} \end{array} \qquad \begin{array}{r} 11011 \\ +10000 \\ \hline 101011 \end{array} = 1x2^5 + 1x2^3 + 1x2^1 + 1x2^0 = 32 + 8 + 2 + 1 = 43$$

$$\begin{array}{r} 27_{16} \\ +\ 16_{16} \\ \hline 3D_{16} \end{array} \qquad \begin{array}{r} 0010\ 0111 \\ +\ 0001\ 0110 \\ \hline \underline{0011}\ \underline{1101} \\ 3\qquad D \end{array}$$

İşaretsiz Sayılarda Çıkarma

- Çıkarma işlemine, sayıların en sağdaki basamaklarından başlanır. Basamaktaki çıkarma işlemi gerçekleşemiyorsa bir üst basamaktan ödünç *elde (taban değeri)* alınarak çıkarma işlemi yapılır. Tüm basamaklarda bu işlem uygulanarak çıkarma işlemi tamamlanmış olur. Kullanılan sayı tabanı, çıkarma işleminde dikkate alınmalıdır.
- İşaretsiz sayılarla çıkarmadaki kural (herhangi bir basamaktaki iki bitin çıkartılması);

0 - 0 = 0

0 - 1 = 1 (Üst basamaktan elde biti ödünç alınmış)

1 - 0 = 1

1 - 1 = 0

- şeklindedir. Sonuç her zaman doğrudur

Çıkarma örnekleri

- Birinci örnekten görüldüğü gibi, ödünç alınan elde biti, işlem basamağında taban, geçtiği basamaklarda ise Taban -1 değerindedir. Ödünç alınan basamaktaki değer taban değeri kadar azalır.

$$\begin{array}{r} 9 \\ -\,3 \\ \hline 6 \end{array} \qquad \begin{array}{ccccc} & \text{3.basamak} & \text{2.basamak} & \text{1.basamak} & \text{0.basamak} \\ & & & 1 & \\ & & 1 & 1 & \\ & 1 & 0 & 0 & 1 \\ - & 0 & 0 & 1 & 1 \\ \hline & 0 & 1 & 1 & 0 \end{array} \qquad \begin{array}{r} 7 \\ -\,3 \\ \hline 04 \end{array} \qquad \begin{array}{r} 0111 \\ -\,0011 \\ \hline 0100 \end{array}$$

$$\begin{array}{r} 35 \\ -\,19 \\ \hline 16 \end{array} \qquad \begin{array}{r} 100011 \\ -\,010011 \\ \hline 010000 \end{array} \qquad \begin{array}{r} 135 \\ -\,076 \\ \hline 059 \end{array} \qquad \begin{array}{r} 10000111 \\ -\,01001100 \\ \hline 00111011 \end{array}$$

Sekiz ve Onaltı tabanlı sayı sistemlerinde çıkarma

$$\begin{array}{r} 35 \\ -\ 17 \\ \hline 16_8 \end{array} \qquad \begin{array}{r} 011101 \\ -\ 001111 \\ \hline \underline{001}\ \underline{110} \\ 1 \quad\ 6 \end{array}$$

$$\begin{array}{r} 67 \\ -\ 51 \\ \hline 16_{10} \end{array} \qquad \begin{array}{r} 1000011 \\ -\ 0110011 \\ \hline 0010000 \end{array} = 1x2^4 = 16$$

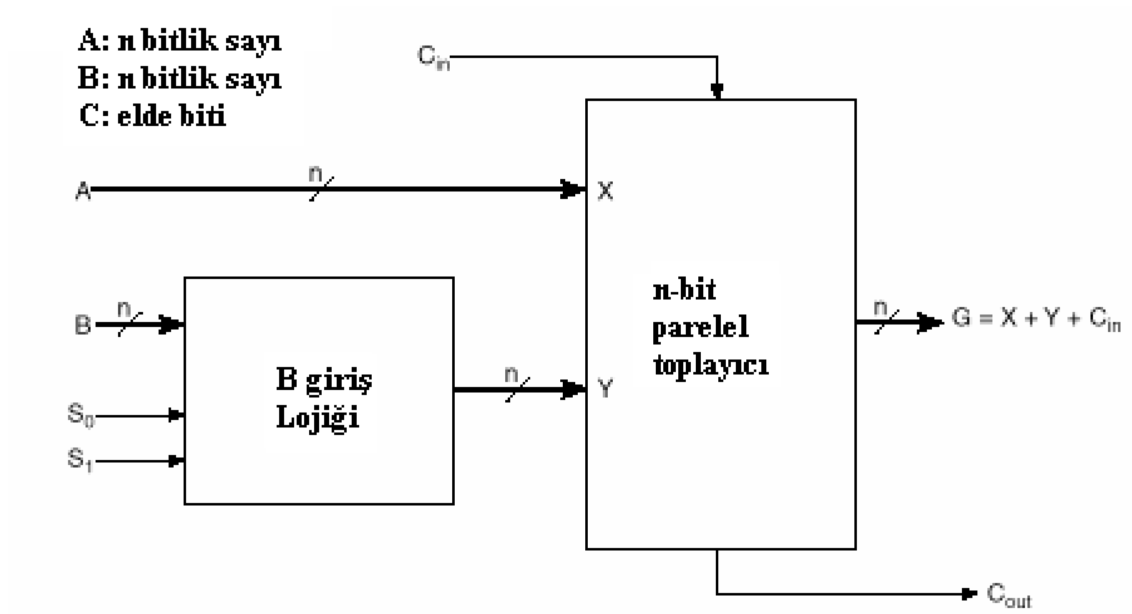
$$\begin{array}{r} 9E \\ -\ 7F \\ \hline 1F_{16} \end{array} \qquad \begin{array}{r} 10011110 \\ -\ 01111111 \\ \hline \underline{0001}\ \underline{1111} \\ 1 \quad\ F \end{array}$$

Tamamlayıcı toplama yoluyla Çıkarma

- Çıkarma işlemini gerçekleştirmek için bir çıkarma donanımsal devresine ihtiyaç vardır. Oysa sadece toplayıcı donanımsal devresi kullanılarak ta çıkarma yapılabilir.
- Bu yöntem de *çıkarılan* aynı kalır. *Çıkan*'ın 2'ye tümleyeni alınıp *çıkarılan* sayı ile toplanır.
- Oluşabilecek *elde* biti göz ardı edilir. Sonuç elde edilmiştir

$$\begin{array}{r} 17 \\ -\ 09 \\ \hline 08 \end{array} \qquad \begin{array}{rl} 10001 & \leftarrow \text{Çıkarılan sayı} \\ -\ 01001 & \leftarrow \text{Çıkan sayı} \\ \hline 01000 & \end{array} \qquad \begin{array}{rl} 10001 & \\ +\ 10110 & \leftarrow \text{Çıkan'ın Taban-1' tümleyeni} \\ \hline (1)00111 & \\ +\ \quad\rightarrow 1 & \\ \hline 01000 & \end{array}$$

Kafa Karıştırma (Aritmetiksel İşlem nasıl yapılır?)



Aritmetik İşlem devresinin Tanım tablosu

Seçme		Giriş	$G = A + Y + C_{in}$	
S_1	S_0	Y	$C_{in} = 0$	$C_{in} = 1$
0	0	all 0's	$G = A$ (transfer)	$G = A + 1$ (increment)
0	1	B	$G = A + B$ (add)	$G = A + B + 1$
1	0	$\overline{B}$	$G = A + \overline{B}$	$G = A + \overline{B} + 1$ (subtract)
1	1	all 1's	$G = A - 1$ (decrement)	$G = A$ (transfer)

Çarpma İşlemi

- Değişik sayı tabanlarında çarpma işlemi alışılagelmiş çarpma kurallarına göre yapılır.
- Aslında çarpma işlemi sola kaydırma ve toplama işleminden başka bir şey değildir. Başka bir deyişle, bir kaydırma ve bir toplama ünitesiyle çarpma işlemi donanımsal olarak gerçekleştirilir.
- Diğer tabanlarda yazılmış sayılar ikilik sisteme dönüştürüldükten sonra iki tabanlı sistem kurallarına göre yapılır.
- **<u>İki tabanlı Sayı Sisteminde Çarpma</u>**

  İki tabanlı sayı sistemindeki çarpma kuralları;

  0 x 0 = 0

  0 x 1 = 0

  1 x 0 = 0

  1 x 1 = 1

Çarpma örneği

- Örnekte olduğu gibi, çarpan sayının işlem basamağındaki bit değeri 1 ise çarpılan 1 bit sola kaydırılarak bir önceki ara sonuç ile toplanır.Çarpanın işlem biti 0 ise bir sonraki ara sonuç iki defa sola kaydırılır.
- Çarpma işleminin sola kaydırma ve toplama işlemi olduğu örnekten de anlaşılmaktadır

$$\begin{array}{r} 7 \\ +3 \\ \hline 21 \end{array}$$

$$\begin{array}{rl} 0111 & \leftarrow \text{Çarpılan} \\ +\,0011 & \leftarrow \text{Çarpan} \\ \hline 0111 & \\ 0111 & \\ 0000 & \\ +\,0000 & \\ \hline 0010101 & \end{array}$$

Sekiz ve Onaltı tabanlı sayılarda çarpma işlemi

Çarpanın her bir basamağının *çarpılanla* çarpılıp elde edilen ara sonuçların birer basamak kaydırılarak toplanmasıyla sonuç elde edilir. Çarpma kuralları kullanılan sayı tabanına göre uygulanmalıdır

```
   16
 x 21
 ----
   16
+ 34
 -----
  356 8
```

```
   37
 x 4C
 ----
  284
+ DC
 -----
 1054 16
```

```
     001110
   x 010001
   --------
     001110
    000000
   000000
  000000
+ 001110
-----------
  011101110
  --- --- ---
   3   5   6
```

```
           00110111
   x       01001100
   ----------------
         0011011100
        00110111000
+    00110111000000
-------------------
      1000001010100
      - ---- ---- ----
      1  0    5    4
```

Bölme İşlemi

- Bölme işlemi de bilinen bölme kuralları uygulanarak gerçekleştirilir. Aslında sağa kaydır çıkar işlemlerinden meydana gelir.
- Çıkarma işlemi de tamamlayıcı toplamayla yapılabildiğinden, sadece bir kaydırıcı ve toplayıcı donanımıyla bölme başarılabilir.
- Değişik tabanlardaki bölme işlemi ikilik sisteme dönüştürüldükten sonra, bu tabanın kurallarıyla yapılır.
- **İki tabanlı sayılarda bölme işlemi**

```
  23 | 2                 10111 | 10
 -2  |----               -10    |-----
 ----| 11                ------ | 1011
  03                       0011
 -02                     - 0010
 ----                    -------
  01                      000011
                         -000010
                         -------
                          000001
```

İşaretli Sayılarda aritmetik işlemler

- Sayısal Sistemlerde iki tabanlı sayı sistemi kullanıldığını biliyoruz. Bu tabanda işaretli sayıların genellikle 2'ye tümlemeyle ifade edildiği de yukarıda açıklamıştı. ,
- Bu bölümde sadece ikilik sistemde 8 bitle ifade edilen sözcüklerden oluşmuş işaretli sayılarla aritmetik işlemlerin nasıl yapıldığı açıklanacaktır.
- Diğer tam kodlamalı sayıların kolaylıkla ikilik sisteme dönüştürüleceği açıklandığından sadece ikilik sistem için açıklamalar yapmak yeterli olacaktır.

İşaretli Sayılarda toplama

- Toplama işleminde 4 değişik durum söz konusu olabilir.
- a)Toplanacak sayılar pozitiftir.
- b) Mutlak değerce büyük olan sayı pozitif diğeri negatif olabilir.
- c) Mutlak değerce küçük olan sayı pozitif diğeri negatif olabilir.
- d)Her iki sayı da negatif olabilir.

$$\begin{array}{r} 14 \\ +\ 07 \\ \hline 21 \end{array} \qquad \begin{array}{r} 00001110 \\ +\ 00000111 \\ \hline 00010101 \end{array}$$

Her iki sayı pozitif ise:
Fazladan elde biti oluşmaz. Sonuç pozitif doğru sayıdır.

$$\begin{array}{r} 93 \\ +\ -40 \\ \hline 53 \end{array} \qquad \begin{array}{r} 01011101 \\ +\ 11011000 \\ \hline 1\ 00110101 \end{array}$$

Pozitif sayı negatif sayıdan büyük ise:
Fazladan elde biti oluşur.
Bu bit gözardı edilerek pozitif sayı elde edilir.

$$\begin{array}{r} 32 \\ +\ -44 \\ \hline -12 \end{array} \qquad \begin{array}{r} 00100000 \\ +\ 11010100 \\ \hline 11110100 \end{array}$$

Negatif sayı pozitif sayıdan büyük ise:
Fazladan elde biti oluşmaz, Sonuç negatif doğru sayıdır.

$$\begin{array}{r} -25 \\ +\ -34 \\ \hline -59 \end{array} \qquad \begin{array}{r} 11100111 \\ +\ 11011110 \\ \hline 1\ 11000101 \end{array}$$

Her iki sayıda negatif ise: Oluşan elde biti atılır.
Kalan doğru negatif sayıdır.

Taşma-Overflow

- Sayısal sistemlerde negatif sayılar 2'ye tümleyen formunda gösterilir. İşaretli sayıların toplanması oldukça basittir. Taşma olmadığı sürece, elde biti oluşursa göz ardı edilir. Sonucun MSB biti işaret bitidir. Sayı buna göre değerlendirilir.

- *Taşma (Overflow): n-bitlik* işaretli sayılarla toplama yapılırken sonuç n-bitten daha fazla çıkabilir.

- Örneğin 8 bitlik (+127, -128 arasındaki sayılar) işaretli sayılardan 127 +126 = 253 sayısı, 8 bitlik işaretli sayı olarak gösterilemez. Yani bu toplama sonucu 9 bit çikar. Buna taşma, overflow vardır denir.
- Toplama sonucunda taşma olup olmadığı, toplanan sayıların ve sonucun işaret bitleri değerlendirilerek anlaşılır.

1.sayı	**2.sayı**	**Sonucun işaret biti**	
Pozitif (İşaret biti 0)	Pozitif (İşaret biti 0)	Negatif (1)	Taşma var
Negatif (İşaret biti1)	Negatif (İşaret biti 1)	Pozitif (0)	Taşma var

Taşma -Örnek

- Yandaki örneklerde sonuçlar 8 bitle ifade edilemez. Taşma vardır.
- **Önemli not**; n bitlik işaretli sayılarla çalışılırken, sonuç sözcüğünün sağdan sola doğru n.biti işaret biti olarak değerlendirilir

$$\begin{array}{r} 126 \\ +125 \\ \hline 251 \end{array}$$

İşaret bitleri 0 (Sayılar pozitif)

$$\begin{array}{r} \mathbf{0}1111110 \\ +\ \mathbf{0}1111101 \\ \hline \mathbf{1}1111011 \end{array}$$

İşaret biti 1

$$\begin{array}{r} -126 \\ +\ -125 \\ \hline -251 \end{array}$$

İşaret bitleri 1 (Sayılar negatif)

$$\begin{array}{r} 10000010 \\ +\ 10000011 \\ \hline \mathbf{1}\ 00000101 \end{array}$$ Sonuç

İşaret biti 0

İşaretli Sayılarda Çıkarma

- **Çıkarma toplama işleminin özel bir halidir.**
- **İşaretli sayılarda çıkarma işlemi, Çıkan sayının 2'ye tümleyeni alınıp çıkarılan sayıyla toplanmasından ibarettir.**
- **Elde biti oluşmuşsa, göz ardı edilmelidir.**
- **İşaretli sayılardaki toplama kuralı geçerlidir.**

7 - 3 = 4

  00000111 Çıkarılan
+11111101 Çıkan
~~1~~00000100 Sonuç

Her iki sayı pozitif. Çıkarılan çıkandan büyük. Çıkan sayının 2'ye tümleyeni alınır, toplama işlemi gerçekleştirilir. Elde biti gözardı edilir.

12 - (-7) = 19

  00001100
+00000111
  00010011

Çıkarılan pozitif. Çıkan negatif, Çıkan sayının 2'ye tümleyeni alınır, toplama işlemi gerçekleştirilir.

-27 - (+18) = -45

  11100101
+ 11101110
~~1~~11010011

Çıkarılan negatif,çıkan pozitif ise, çıkan sayının 2'ye tümleyeni alınır, toplama işlemi gerçekleştirilir. Elde biti gözardı edilir.

-118 - (-92) = -26

  10001010
+01011100
  11100110

Çıkarılan ve çıkan sayılar negatif ise, çıkan sayının 2'ye tümleyeni alınır,toplama gerçekleştirilir.

İşaretli Sayılarda Çarpma

- İki tabanındaki sayılar çarpılırken gerçek (tümlenmemiş) durumda olmalıdırlar. Çarpma sonucunun (çarpım) işareti çarpan ve çarpılanın işaretlerine bağlıdır[R].
- İşaretler aynı ise çarpım pozitiftir. İşaretler farklı ise çarpım negatiftir.
- Aşağıdaki örnekte **01010111 (+87)** ile **11000111 (-57)** işaretli sayılarının çarpılması verilmiştir. Çarpma işleminde işaret bitleri atılır ve -57'nin 2'ye tümleyeni **(0111001)** alınarak çarpma işlemi, 2'lik tabanda pozitif sayıların çarpılması kurallarına göre gerçekleştirilir.
- Çarpılan ve çarpan sayılarının işaret bitlerinden, **sonucun** negatif çıkacağı görülür.
- Bunun için sonucun 2'ye tümleyeni alınarak gerçek sonuç elde edilir (Sonuca işaret biti eklenerek).
- Buna göre; 1 **0110010100001 (-4959**) elde edilir.

7bit 6bit 5bit 4bit 3bit 2bit 1bit

1010111	Çarpılan	**87**
x 0111001	Çarpan	**57**
1010111	1.bitin çarpımı	
+00000000	2.bitin çarpımı	
001010111	Ara sonuç	
+ 0000000	3.bitin çarpımı	
0001010111	Ara sonuç	
+1010111	4.bitin çarpımı	
1100001111	Ara sonuç	
+1010111	5.bitin çarpımı	
100001111111	Ara sonuç	
+1010111	6.bitin çarpımı	
1001101011111	Ara sonuç	
+0000000	7.bitin çarpımı	
1001101011111	SONUÇ	**4959**

İşaretli Sayılarda Bölme

- İkilik sayılar bölünürken her iki sayı da gerçek (tümlenmemis) durumda olmalıdır. Yani işaretsiz sayılarla bölme işlemi yapılır.
- Bölme işlemi yapılırken şu sıra izlenir;[R]

1-Bölen ve bölünen sayıların işaretlerinin aynı olup olmadığına bakılarak sonucun işareti belirlenir.

2-Bölen bölünenden çıkarılıp (Çıkarma işlemi genellikle Tamamlayıcı toplama metoduna görede yapılır) ilk *kısmi kalan* bulunur ve bölüme 1 eklenir. Eğer bu *kısmi kalan, bölenden* büyükse (tamamlayıcı toplama yoluyla çıkarma yapıldığında pozitif ise) 3. asamaya geçilir.

Eğer sonuç *bölenden* küçükse, sıfırsa (tamamlayıcı toplama yoluyla çıkarma yapıldığında negatif ise) bölme tamamlanmıştır.

3-Bölen kısmi kalandan çıkarılarak bölüme 1 eklenir. Eger sonuç pozitifse işlem sürdürülür. Sonuç sıfır ya da negatifse bölme tamamlanmıştır.

4- Bölüm(Sonuç)'un işareti aşağıdaki şekilde belirlenir.

- a-) Bölen ve bölünen aynı işaretli ise bölüm pozitiftir. İşaret biti konarak sonuç oluşturulur.
- b) Bölen ve bölünen farklı işaretli iseler, bölüm negatiftir. Bölüm'ün 2'ye tümleyeni alınarak sonuç negatif olarak elde edilir.

- Örnek olarak; 103 / 26 = ? işleminin nasıl yapılacağı aşağıda verilmiştir.

Bölme örnekleri (103/26 =?)

	Bölünen	Bölen		
	01100111	00011010	Bölüm : Başlangıçta 00	
	− 00011010	11 Bölüm	1.çıkarma işleminde bölüme 1 eklenir	01
1.kismi kalan	1001101			
	− 0011010		1.kısmi kalan − Bölen işlemi, bölüme'e 1 eklenir.	10
2.kismi kalan	0110011			
	− 0011010		2.kismi kalan − Bölen işlemi, bölüm'e 1 eklenir.	11
	011001	Kalan	Kalan bölümden küçük, bölme işlemi bitmiştir.	

Bölüm : 11
Kalan: 011001

	Bölünen	Bölen		
	01100111	00011010	Bölüm : Başlangıçta 00	
	+ 11100110	11 Bölüm	1.çıkarma işleminde bölüme 1 eklenir	01
1.kismi kalan	~~1~~01001101			
	+ 11100110			
2.kismi kalan	~~1~~00110011		1.kısmi kalan + Bölen'in 2'ye tümleyeni: pozitif, bölüm'e 1 eklenir.	10
	+ 11100110			
3.kısmi kalan (kalan)	~~1~~00011001		2.kısmi kalan + Bölen'in 2'ye tümleyeni: pozitif, bölüm'e 1 eklenir.	11
	+ 11100110			
	11111111		3.kısmi kalan + Bölen'in 2'ye tümleyeni : negatif, bölme işlemi bitmiştir.	

Bölüm : 11
Kalan: 011001