

Sayı Sistemleri

Geçen Hafta

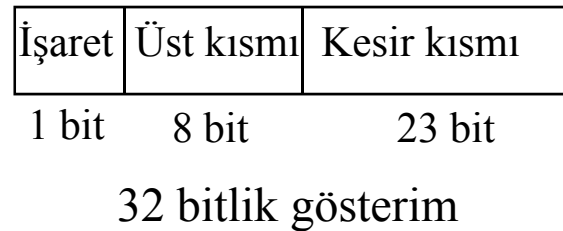
- Analog ve Sayısal Büyüklük Kavramı
- Sayı Sistemleri
- İkilik Sistemde Aritmetik İşlemler
- İkili Sayıların 1'e ve 2'ye Tümleyeninin Bulunması
- İşaretli Sayılar

Bu Hafta

- Kayan Noktalı Sayılar
- İşaretli Sayılarda Aritmetik İşlemler
 - Toplama ve Çıkarma İşlemleri
 - Çarpma ve Bölme İşlemleri

Kayan Noktalı Sayılar

Diğer gösterim şekilleri kullanılarak tüm tamsayı değerlerini göstermek oldukça fazla sayıda bit gerektirecektir. Bunun yanı sıra kesirli sayıların da gösterimi gereklidir. Kayan nokta gösterimiyle, hem çok küçük hem de çok büyük sayıları bilimsel formda göstermek mümkündür. Örneğin IEEE-754'e göre kayan noktalı sayılar 32 bit ile gösterilir; bir bit işareti, 8 bit üst kısmını, 23 bit de kesirli kısmı gösterir. 64 bitlik ve 80 bitlik gösterimler de mevcuttur.



Normalizasyon

Herhangi ikili bir sayı bilimsel notasyonda farklı şekillerde gösterilebilir. Bir standart getirmek gayesiyle en soldaki bit değeri 1 olacak şekilde düzenlemeye gidilir. Buna normalizasyon denir.

Örneğin 10011001011_2 sayısı $1.0011001011 \times 2^{10}$ olarak gösterilir.

Normalize edilmiş bu formatın kayan nokta gösteriminde, üst kısmı biaslı olarak kullanılır yani gerçek üst değerine $127 (2^{8-1} - 1)$ sayısı eklenir.

Bunun sebebi ise çok büyük ve çok küçük sayıların gösterimine olanak sağlamaktır. Gerçek üst değeri -126 ile 128 arasındadır.

Kayan nokta gösteriminde tüm bitlerin sıfır olması 0 olarak, üst kısmındaki bitlerin hepsinin 1, kesir kısmındaki tüm bitlerin 0 olması da sonsuz olarak yorumlanır.

Kayan Noktalı Sayılar

Örnek: $1.0011001011 \times 2^{10}$ sayısının kayan nokta gösterimi;

Sayı pozitif olduğundan işaret biti 0'dır. Üst kısmı $10+127$ (bias)=137 ve kesirli kısım da 0011001011_2 dir.

0	10001001	00110010110...0
1 bit	8 bit	23 bit

Yine aynı sayının negatifi gösterilmek istenirse, işaret bitinin 1 yapılması yeterlidir.

32 bitlik kayan nokta gösterimi ile gösterilebilecek en büyük sayıyı ikilik sistemde gösterebilmek için 129 bite ihtiyaç duyulacaktı.

Kayan Noktalı Sayılar

Örnek: Sayıların 4 bit ile gösterildiği bir sistemde, işaret kısmına 1 bit, üst kısmına 2 bit ve kesir kısmına da 1 bit ayırdığımızı düşünelim. Normalize edilmiş formata göre gösterebileceğimiz sayı aralığını inceleyelim.

İşaret	Üst kısım	Kesir kısmı
1 bit	2 bit	1 bit

Bias değerimiz $2^{2-1}-1=1$ 'dir.

Yazılabilecek sayılar;

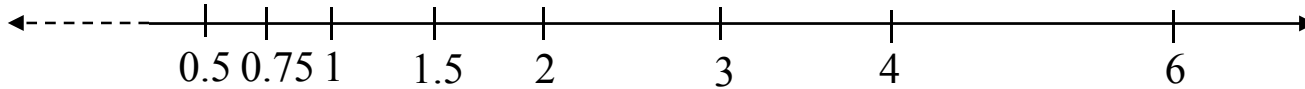
$$1.0 \times 2^{-1} = 0.5 \quad 1.1 \times 2^{-1} = 0.75$$

$$1.0 \times 2^0 = 1 \quad 1.1 \times 2^0 = 1.5$$

$$1.0 \times 2^1 = 2 \quad 1.1 \times 2^1 = 3$$

$$1.0 \times 2^2 = 4 \quad 1.1 \times 2^2 = 6$$

Örnek: (Devamı)



- Sayı ekseninde sayıların eksi değerleri de vardır. 4 bit ile gösterilebilecek sayı adedi 16 olduğundan kayan nokta formatıyla da 16 adet sayı gösterilebilmektedir. Ancak hassasiyet artmıştır ve sayılar ekseninde homojen olarak dağılmamıştır.
- Dikkat edilecek olursa 0.5 ile 1 arasında 3 sayı gösterebilmekteyiz. Aynı kural 1 ile 2 arasındaki, 2 ile 4 arasındaki sayılar için de geçerlidir.
- Şayet 2 biti kesir kısmı, 1 biti üst kısmı için ayırmış olsaydık, daha da hassas olarak sayıları gösterebilirdik (1, 1.25, 1.5, 1.75, 2, 2.5, 3, 3.5). Ancak, üst kısmına daha az bit ayırdığımızdan gösterebileceğimiz sayının büyüklüğü azalmıştır.

İşaretili Sayılarda Aritmetik İşlemler

Bu kısımda daha sıklıkla kullanılan 2'ye tümleyen sistemi esas alınacaktır.

Toplama ve çıkarma işlemleri:

- İki pozitif sayının toplamı pozitiftir.

Örnek: 00001001 (9)
+ 00000100 (4)

00001101 (13)

- Pozitif bir sayıyla daha küçük negatif bir sayının toplamı pozitiftir. Elde biti oluşursa göz ardı edilir.

Örnek: 00001001 (9)
+ 11111010 (-6)

100000011 (3)

İşaretili Sayılarda Aritmetik İşlemler

- Pozitif sayının büyüklüğünden daha büyük negatif bir sayı olursa, oluşan toplam 2'ye tümleyen formunda negatif bir sayıdır.

Örnek: 00001000 (8)

+ 11110100 (-12)

11111100 (2'te tümleyen formunda -4'e karşılık gelir)

- İki negatif sayının toplamı negatiftir, elde biti oluşursa göz ardı edilir.

Örnek: 11111101 (-3)

+ 11111110 (-2)

111111011 (-5)

İşaretili Sayılarda Aritmetik İşlemler

- İşaretili sayılar toplandığında elde biti oluşursa göz ardı edilir. İki istisnai durum vardır; ilki, iki sayı da pozitif ancak işaret biti 1 ise, diğeri ise iki sayı da negatif ancak işaret biti 0 ise durumlarıdır. Her iki durumda da bir taşma meydana gelmiştir yani toplamı ifade etmek için 1 bite daha gereksinim olduğu söylenebilir.

Örnek:
$$\begin{array}{r} 01101110 \quad (110) \\ + 00011001 \quad (25) \\ \hline 10000111 \quad (\text{taşma durumu}) \end{array}$$

İşaretili sayılarda çıkartma işlemi aslında toplama işleminin değişik bir formudur. Yani çıkartılan sayının 2'ye tümleyeni ile çıkan sayının toplamından sonuç elde edilir.

Örnek: 8-3 işlemi $8+(-3)$ olarak da yazılabilir.

$$\begin{array}{r} 00001000 \quad (8) \\ + 11111101 \quad (-3) \\ \hline \underline{100000101} \quad (5) \end{array}$$

Çarpma ve Bölme İşlemleri

Çarpma, toplama işlemiyle gerçekleştirilebilir. Çarpılan sayının, çarpanın değeri kadar kendi üzerine toplanması esasına dayanır. Yani 4×2 işlemi $4+4$ şeklinde düşünülür. Çarpan değerinin büyük olması durumunda zahmetli bir hal alır bu yüzden kısmi çarpım metodu kullanılır;

- İlk olarak çarpılan ve çarpanın aynı işarete sahip olup olmadığına bakılır ve çarpım sonucunun işareti elde edilir.
- Negatif sayının pozitif değeri alınır.
- Çarpanın en az anlamlı bitinden başlanır ve kısmi çarpımlar elde edilir. Şayet çarpan 1 ise çarpılanın kendisi, 0 ise 0 değerinden kısmi çarpım elde edilir. Her kısmi çarpım bir sola kaydırılır.
- Ard arda gelen kısmi çarpımlar toplanır ve ara toplamalar elde edilir.
- İlk adımda işaret biti negatifse, sonucun ikiye tümleyenini alınır. Pozitifse olduğu gibi bırakılır. Elde edilen çarpımın önüne işaret biti eklenir.

Çarpma ve Bölme İşlemleri

Örnek: $25 \times (-4)$ işlemini yapalım;

11001	
$\times 100$	-4'ün pozitif değeri
<u>00000</u>	1.kısmi çarpım
$+ 00000$	2.kısmi çarpım
<u>000000</u>	ara toplam
$+ 11001$	3. kısmi çarpım
<u>1100100</u>	sonuç

Çarpımın sonucu negatif olacağından sonucun ikiye tümleyeni alınır;

1100100_2 'ın ikiye tümleyeni 0011100_2 ve işaret biti eklendiğinde 10011100_2 elde edilir.

Bölme İşlemi

Bölme, çıkarma işlemiyle gerçekleştirilebilir. Çıkarma, toplama işlemi ile yapılabildiğinden bölme işlemi de toplama işlemiyle yapılabilir. Bölenin bölünenden çıkarılma sayısı bölümü verecektir. Bölme işleminde şu adımlar takip edilir;

- İlk olarak bölünen ve bölenin aynı işarete sahip olup olmadığına bakılır ve bölme sonucunun işareti elde edilir.
- Her iki sayının da pozitif formu elde edilir.
- Bölüme 0 değeri atanır.
- Bölünenden bölen çıkartılır (2'ye tümleyen ile toplama işlemine dönüştürülebilir), kısmi kalan bulunur ve bölüme 1 eklenir. Kısmi kalan pozitifse bir sonraki adıma gidilir, 0 veya negatifse bölme işlemi tamamlanmıştır.
- Kısmi kalandan bölen çıkartılır ve bölüme 1 eklenir. Sonuç pozitifse sonraki kısmi kalan için işlemler tekrar edilir. Sonuç 0 veya negatifse bölme işlemi tamamlanmıştır.

Bölme İşlemi

Örnek: 25/6 işlemini yapalım.

Her iki sayı da pozitif olduğundan sonuç pozitiftir.

$$25 = 00011001_2 \text{ ve } 6 = 00000110_2$$

Bölüme 0 değeri atanır. Bölünenden bölen çıkartılır;

$$\begin{array}{r} 00011001 \\ + 11111010 \text{ (6'nın 2'ye tümleyeni)} \\ \hline 100010011 \text{ (elde biti atılır) kısmi kalan pozitif olduğundan} \end{array}$$

bölüm=0+1=1

Kısmi kalandan bölen çıkartılır;

$$\begin{array}{r} 00010011 \\ + 11111010 \\ \hline 100001101 \text{ (elde biti atılır) kısmi kalan pozitif olduğundan} \end{array}$$

bölüm=1+1=2

Kısmi kalandan bölen çıkartılır;

$$\begin{array}{r} 00001101 \\ + 11111010 \\ \hline 100000111 \text{ (elde biti atılır) kısmi kalan pozitif olduğundan} \end{array}$$

bölüm=2+1=3

Örnek: (Devamı)

Kısmi kalandan bölen çıkartılır;

$$\begin{array}{r} 00000111 \\ + 11111010 \\ \hline 100000001 \end{array} \quad (\text{elde biti atılır) kısmi kalan pozitif olduğundan}$$

bölüm=3+1=4

Kısmi kalandan bölen çıkartılır;

$$\begin{array}{r} 00000001 \\ + 11111010 \\ \hline 11111011 \end{array} \quad \text{elde oluşmadığından sonuç negatiftir, bölme işlemi}$$

tamamlanmıştır. Bölme işleminin sonucu bölümün aldığı değerdir,

4'ü ikilik sistemde ifade edersek 00000100_2 dır. Kalan da bir önceki işlemde 1 olarak bulunur.