İşaretli Tamsayı Gösterimi

1. İşaretli Büyüklük

Bir işaretli büyüklük sayısında en soldaki basamak bir işaret içerir. Diğer basamaklarda ise sayısal değerin büyüklüğü (mutlak değeri) gösterilir.

Örnek
8 basamaklı bir sözcükte, -1 değeri 10000001 şeklinde, +1 değeri ise 00000001 şeklinde gösterilir.

Soru
İşaretli-büyüklük metodu ile N basamakta gösterilebilecek en büyük ve en küçük işaretli tamsayı nedir? Formülü yazınız.

Cevap :
En küçük: $-2^{(N-1)} - 1$
En büyük: $2^{(N-1)} - 1$.

Toplama:
Bu gösterimle ifade edilen tamsayılarla yapılan aritmetik hesaplamalarda

(1) Eğer işaretler aynı ise, büyüklükler toplanır ve sonuç için aynı işaret kullanılır;
(2) Eğer işaretler farklı ise, büyük olan sayı belirlenir. Sonucun işareti büyük olan sayının işareti ile aynıdır, büyüklüğü ise küçük olan sayıdan büyük olan sayı çıkarılarak bulunur.

Soru
İşaretli-büyüklük metodunu kullanarak 01001111_2 ile 00100011_2 'i toplayınız.

Cevap:

		1 1 1 1	⇐ carries
0		1 0 0 1 1 1 1	(79)
0	+	0 1 0 0 0 1 1	+ (35)
0		1 1 1 0 0 1 0	(114)

Aritmetik hesaplama, sağdan yedinci basamağa ulaşana dek tıpkı ondalık toplama işlemi gibi ilerler (eldeler dahil). Bu aşamada bir elde varsa taşma durumu ortaya çıkmış demektir, elde atılır. Bu da toplama işleminin yanlış sonuç vermesine sebep olur. Bu örnekte taşma durumu yoktur.

İşaretli-büyüklükte, işaret basamağı sadece işaret için kullanılır. Dolayısıyla bu basamakta elde işlem görmez. Eğer yedinci basamaktan gönderilen bir elde varsa, sonuç yedinci basamak taşması olarak adlandırılır ve yanlış toplamı verir.

Soru
İşaretli-büyüklük aritmetiğini kullanarak 01001111_2 ile 01100011_2'i toplayınız.

Last carry	1 ←	1 1 1 1	⇐ carries
overflows and is	0	1 0 0 1 1 1 1	(79)
discarded.	0 +	1 1 0 0 0 1 1	+ (99)
	0	0 1 1 0 0 1 0	(50)

Yandaki hatalı sonuç elde edilir: 79 + 99 = 50.

Çıkarma:

Toplama işleminde olduğu gibi, işaret-büyüklüğü çıkarma işlemi de kalem-kağıt ondalık aritmetiği şeklinde yapılır. Bazen *çıkarılan* sayının basamaklarından ödünç almak gerekebilir.

Soru
İşaretli-büyüklük aritmetiğini kullanarak 01001111_2 'den 01100011_2 'i çıkarınız.

			0	1	1	2			⇐ borrows
0		1	~~1~~	~~0~~	~~0~~	~~0~~	1	1	(99)
0	−	1	0	0	1	1	1	1	− (79)
0		0	0	1	0	1	0	0	(20)

İşaret-büyüklük gösteriminde $01100011_2 - 01001111_2 = 00010100_2$ olduğunu buluruz.

Soru
İşaretli-büyüklük aritmetiğini kullanarak 01100011_2 (99)'den 01001111_2 (79)'i çıkarınız.

Çıkan çıkarılandan daha büyük olduğundan, tüm yapmamız gereken farkın işaretini değiştirmek. Böylece işaret-büyüklük gösteriminde $01001111_2 - 01100011_2 = 10010100_2$ olduğunu buluruz.

Eğer iki sayı birbirinden farklı işaretlere sahipse ne yapmalıdır?

Soru
İşaretli-büyüklük aritmetiğini kullanarak 100100112 (–19) ile 000011012 (+13)'yi toplayınız.

				0	1	2			⇐ borrows
1		0	0	~~1~~	~~0~~	~~0~~	1	1	(−19)
0	−	0	0	0	1	1	0	1	+ (13)
1		0	0	0	0	1	1	0	(−6)

İki sayı arasından büyük olanı belirlenir. Bu sayının işareti sonucun işareti olacaktır.

Tümleyen Sistemleri

İşaretli büyüklükte sıfır için iki gösterim vardır, 10000000 ve 00000000. Mantık karmaşıktır. Bundan dolayı daha basit metodlar gereklidir.

Bir ondalık sayıdan diğerini nasıl çıkarabiliriz?

Bir ondalık sayıdan diğerini çıkarabilmek için bu sayıya tüm dokuzlardan çıkanın farkını ve bu işlemde bulduğumuz eldeyi eklemek gerekir. Buna çıkanın dokuzlu tümleyenini alma, ya da daha resmi olarak, çıkanın *azalan taban tümleyenini* bulma denir.

Örnek
167 - 52 işleminin sonucunu bulmaya çalışalım.

999'un 52'den farkını alırız, 947'yi buluruz. Dokuzlu tümleyen aritmetiğinde, 167 - 52 = 167 + 947 = 114 buluruz. Yüzler basamağından gelen eldeyi birler basamağına geri ekleriz, böylece 167 - 52 = 115 doğru sonucuna ulaşırız.

Tümleyen sistemlerinin işaretli büyüklük metoduna göre avantajı, işaret basamaklarının ayrı ayrı işlem görmemesidir. Ancak yine de bir sayının işaretini en soldaki basamağına bakarak kolaylıkla anlayabiliriz.

Tümleyen gösterimi çıkarma işlemini toplama işlemine çevirerek basitleştirir. Bununla birlikte negatif sayıları da göstermemize olanak sağlar.

10 tabanında bir sayının azalan taban tümleyenini bulmak için, taban-1'den (onluk sistemde 9) çıkanı çıkarılır.

Örneğin 2468'in dokuzlu tümleyeni 9999 – 2468 = 7531 'dir. İkilik sistemde buna benzer bir işlem için, taban-1'den yani 1'den çıkarırız. Örneğin 0101_2'nin birli tümleyeni $1111_2 – 0101 = 1010$ 'dur.

Örnek:
Birli tümleyen aritmetiğini kullanarak 23_{10} ile $–9_{10}$ sayılarını toplayınız.

	1 ← 1 1 1 1 1	⇐ carries
	0 0 0 1 0 1 1 1	(23)
The last	+ 1 1 1 1 0 1 1 0	+(–9)
carry is added	0 0 0 0 1 1 0 1	
	+ 1	
to the sum.	0 0 0 0 1 1 1 0	14_{10}

En soldaki basamaktan elde 1 ya da 0 kalır, bu da toplamın en sağdaki basamağına eklenir. (Buna *end carry-around* denir ve azalan taban tümleyeni kullanılarak bulunur.)

Soru:
Birli tümleyen aritmetiğini kullanarak 9_{10} ile -23_{10} sayılarını toplayınız.

$$
\begin{array}{lrr}
\text{The last} & 0 \leftarrow \;\; 0\;0\;0\;0\;1\;0\;0\;1 & (9) \\
\text{carry is zero} & \underline{+\;1\;1\;1\;0\;1\;0\;0\;0} & \underline{+(-23)} \\
\text{so we are done.} & 1\;1\;1\;1\;0\;0\;0\;1 & -14_{10}
\end{array}
$$

Birli tümleyenin en önemli dezavantajı, sıfır için hala iki gösterime sahip olmasıdır: 00000000 ve 11111111. Bu ve diğer sebeplerden dolayı, bilgisayar mühendisleri uzun yıllar önce ikili sayılar için birli tümleyeni kullanmaktan vazgeçtiler ve daha verimli olan ikili tümleyen gösterimini kullanmayı tercih ettiler.

İkili tümleyen

İkili tümleyen taban tümleyeninin bir örneğidir. -2 onlu tümleyende 1000-2 = 998 şeklinde gösterilir.

İkili tümleyen birli tümleyenin 1 arttrılmasından ibarettir.

Çıkan (tümleyen ve eklenen sayı) başlangıçta arttırıldığından end carry-around olmayacaktır. En soldaki basamaklar dahil tüm eldeler atılır.

Örnek:
İkili tümleyen aritmetiğini kullanarak 9_{10} ile -23_{10} sayılarını toplayınız.

$$
\begin{array}{rr}
0\;0\;0\;0\;1\;0\;0\;1 & (9) \\
+\;\underline{1\;1\;1\;0\;1\;0\;0\;1} & \underline{+(-23)} \\
1\;1\;1\;1\;0\;0\;1\;0 & -14_{10}
\end{array}
$$

İkili tümleyen gösterimine göre 111100102 sayısının –1410 sayısına eşit olduğunu kanıtlayınız!

Taşma

İki pozitif sayı toplanır ve sonuç negatif olursa, veya iki negatif sayı toplanır ve sonuç pozitif olursa taşma oluşur. İkili tümleyen gösteriminde bir pozitif ve bir negatif sayı toplanırsa taşma oluşmaz.

Taşma Durumunun Anlaşılması İçin Basit Kural: *Eğer işaret basamağındaki elde basamak dışındaki eldeye eşitse taşma oluşmaz, eşit değilse taşma (dolayısıyla hata) oluşur.*

Örnek:

İkili tümleyen aritmetiğini kullanarak 126_{10} ve 8_{10} sayılarının ikili sistemde toplamını bulunuz.

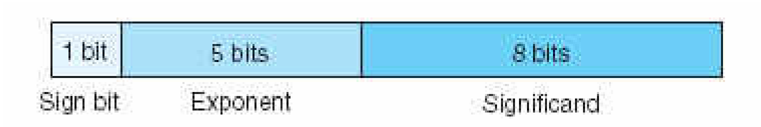
	0←	1 1 1 1	⇐ carries
Discard last		0 1 1 1 1 1 1 0	(126)
carry.	+	0 0 0 0 1 0 0 0	+(8)
		1 0 0 0 0 1 1 0	(−122???)

Kayan Noktalı Gösterim

Bilimsel gösterimde sayılar iki parçada ifade edilir: kesirli parça *mantissa* ve mantissanın alacağı değere 10'un kuvvetini gösteren bir üstel parça. Böylece 32.767 sayısını bilimsel gösterimde ifade edebilmek için 3.2767×10^4 yazılabilir.

Basit Bir Model

Kayan nokta sayılar üç parçadan oluşur: bir işaret basamağı, bir üst parçası (2'nin kuvvetinde üsteli gösteren) ve *significand* (mantissa yerine kullanılmaktadır) adında bir kesirli parça.



Modelimizde 17 ondalık sayısını tutmak istediğimizi düşünelim.
$17 = 17.0 \times 10^0 = 1.7 \times 10^1 = 0.17 \times 10^2$ olduğunu biliyoruz.
Benzer şekilde, ikili sistemde de şöyledir:
$17_{10} = 10001_2 \times 2^0 = 1000.1_2 \times 2^1 = 100.01_2 \times 2^2 = 10.001_2 \times 2^3 = 1.0001_2 \times 2^4 = 0.10001_2 \times 2^5$.

Eğer bu son şekli kullanırsak, kesirli parçamız 10001000 ve üstümüz 00101 olacak:

0	0	0	1	0	1	1	0	0	0	1	0	0	0

Eğer bu modelde $65536 = 0.1_2 \times 2^{17}$ olduğunu göstermek istersek:

0	1	0	0	0	1	1	0	0	0	0	0	0	0

Negatif bileşenlerle sorunlar?

Eğer 0.25 sayısını tutmak istersek, bu şekilde yapmamız mümkün değil. Çünkü 0.25 sayısı 2^{-2} ifadesine eşittir ve üst –2 gösterilemez. Bu problemi üste bir işaret basamağı ekleyerek çözebilirdik, ancak bu durumda saptırılmış bir üst kullanmak daha verimli olacaktır.

İstenilen üst aralığındaki tam sayılar ilk olarak her bir üste bu sabit sapma değerinin eklenmesiyle ayarlanır.

Bu durumda 16 sayısını seçebilirdik. Üst alanında 16'dan büyük herhangi bir sayı pozitif bir değeri gösterecektir. 16'dan küçük değerler ise negatif değerleri gösterecektir. Buna 16-fazlalık gösterimi denir, çünkü üstün gerçek değerini bulabilmek için 16 çıkarmamız gerekir.

17 sayısını tuttuğumuz örneğe geri dönecek olursak, $17_{10} = 0.10001_2 \times 2^5$ olduğunu hesaplamıştık. Saptırılmış üst şimdi 16 + 5 = 21 oldu:

0	1 0 1 0 1	1 0 0 0 1 0 0 0

Eğer $0.25 = 1.0 \times 2^{-2}$ sayısını tutmak isteseydik, aşağıdaki gibi olurdu:

0	0 1 1 1 0	1 0 0 0 0 0 0 0

Her sayının benzersiz gösterimi: (Normalizasyon)

Bu sistemle ilgili oldukça büyük bir problem daha var: Her sayı için benzersiz bir gösterim yapamıyoruz. Aşağıdakilerin tümü birbirine eşittir:

0	1 0 1 0 1	1 0 0 0 1 0 0 0	=
0	1 0 1 1 0	0 1 0 0 0 1 0 0	=
0	1 0 1 1 1	0 0 1 0 0 0 1 0	=
0	1 1 0 0 0	0 0 0 1 0 0 0 1	

Çünkü bunlar gibi eş anlamlı biçimler dijital bilgisayarlar için pek uygun değildir. En soldaki basamağın her zaman 1 olduğu bir kural getirilmiştir. Buna *normalizasyon* denir. Bu kuralın diğer bir avantajı olarak, 1 sonuca etki eden ekstra bir doğruluk basamağı olarak significand içinde yer alır.

Örnek:

0.03125_{10} sayısını normalize edilmiş 16-fazlalık sapmalı kayan nokta biçiminde ifade ediniz.

Cevap:

$0.03125_{10} = 0.00001_2 \times 2^0 = 0.0001 \times 2^{-1} = 0.001 \times 2^{-2} = 0.01 \times 2^{-3} = 0.1 \times 2^{-4}$.

Sapma uygulandığında üst alanı 16 - 4 = 12 olur.

0	0 1 1 0 0	1 0 0 0 0 0 0 0

Bu örnekte sayıyı 1'i kapsayan normalizasyon gösterimini kullanarak ifade etmediğimize dikkat ediniz.

Kayan Nokta Aritmetiği

Aşağıdaki normalize edilmiş 16 sapmalı 14 basamakta gösterilen ikili sayıları toplayınız.

0	1	0	0	1	0	1	1	0	0	1	0	0	0	+

0	1	0	0	0	0	1	0	0	1	1	0	1	0

$$
\begin{array}{r}
11.001000 \\
\underline{+\,0.10011010} \\
11.10111010
\end{array}
$$

Yeniden normalize edersek büyük üstü elde ederiz sağdaki basamağı kısaltırız. Böylece aşağıdaki sonucunu elde ederiz.:

0	1	0	0	1	0	1	1	1	0	1	1	1	0

Kayan Nokta Hataları

128.5 sayısını ikiliye çevirirsek 9 basamak uzunluğundaki 10000000.1 sayısını elde ederiz. Ancak significand değerimiz sadece 8 basamak tutabiliyor. Genellikle en sağdaki basamak atılır veya bir sonraki basamağa yuvarlanır. Bunu nasıl gerçekleştirdiğimiz çok önemli değil, ancak sistemimizde bir hata ile karşılaştık.

We can compute the relative error in our representation by taking the ratio of the absolute Gösterimimizdeki ilişkili hatayı hatanın mutlak değerinin sayının gerçek değerine oranını alarak hesaplayabiliriz 128.5 örneğini kullanarak aşağıdaki sonuca ulaşabiliriz:

$$\frac{128.5 - 128}{128} = 0.003906 \approx 0.39\%.$$