Kayan Noktalı Sayılar (*Floating Point Numbers*)

10 Tabanı:

$(12.34)_{10} = 12 + \frac{34}{100}$ ya da $(12.34)_{10} = 12 + \frac{3}{10} + \frac{4}{100} = 1 \cdot 10^1 + 2 \cdot 10^0 + 3 \cdot 10^{-1} + 4 \cdot 10^{-2}$

2 Tabanı:

$(101.11)_2 = 5 + \frac{3}{4}$ ya da $(101.11)_2 = 5 + \frac{1}{2} + \frac{1}{4}$, $(0.1111)_2 = \frac{15}{16}$, $(0.1)_2 = \frac{1}{2}$

Virgüllü (noktalı) sayıları bellekte tutmak için akla ilk gelen yöntem sabit noktalı (*fixed radix*) gösterilimdir. Sayının noktadan önceki ve sonraki kısımları için sabit uzunlukta yerler ayrılır.

Sabit noktalı gösterilim pratik değildir. Bellekte fazla yer kaplar, işlem yapmak için uygun değildir.

Örneğin 1 trilyon (10^{12}) göstermek için 40 bit gerekir. $10^{12} \approx 2^{40}$

Benzer şekilde virgülden sonra $1/10^{12}$ hassasiyet için de 40 bit gereklidir.

Toplam 80 bit.

Üstel gösterilim (***Scientific notation, exponential notation***) kullanılır:

$\pm F \cdot B^{\pm E}$

F: *Fraction* (Kesir, Mantis)

E: *Exponent* (Üs)

B: *Base* (Taban)

Bellekte: ± , **F** ve **E** tutulur.

Örnek:

976,000,000,000,000 = 0.976×10^{15}

0.000 000 000 000 976 = 0.976×10^{-12}

Normalize Sayı:

Noktanın yerine önceden karar verilir ve bu yer bilgisi bellekte tutulmaz.

Örneğin, noktanın her zaman sıfırdan farklı en yüksek anlamlı sayının solunda olduğu kabul edilir.

3.14 = 0.314×10^1

Örneğin bellekte ±FFF ±EE şeklinde tutulabilir. +314+01

Yükseltilmiş Üs (*Biased Exponenet*):

Üs değerinin negatif olmaması için üs değeri bellekte saklanmadan önce belli bir değer "ökçe" (*bias*) ile toplanır (üs yükseltilir).

Böylece üssün işaretinin saklanmasına gerek kalmaz ve aritmetik işlemlerde (karşılaştırmada) kolaylık sağlanır.

IEEE 754 Standardı (1985, güncelleme 2008)

Single (32 bit)

İşaret	E	F
S	Üs	Kesir
1	8	23

Üs 127 yükseltilmiştir

Double (64 bit)

İşaret	E	F
S	Üs	Kesir
1	11	52

Üs 1023 yükseltilmiştir

Güncel standartta 16 bitlik (*half*) ve 128 bitlik (*quadruple*) sayılar da bulunmaktadır.

E'deki bit sayısı k olmak üzere üs $(2^{k-1} - 1)$ kadar yükseltilir.

Normalize Sayı (IEEE 754):

Noktanın her zaman sıfırdan farklı en yüksek anlamlı sayının sağında olduğu kabul edilir.

İkili düzende çalışıldığına göre "0"dan farklı sayı "1"dir.

$(10110.101)_2 = 1.0110101 \times 2^4$

Noktadan önce her zaman 1 olduğu bilindiğinden bu 1 değeri de bellekte tutulmaz. Buna **gizli 1** (*hidden one*) denir.

Örnek:

$(+22.625)_{10}$?

$(22)_{10} = (10110)_2$

.625'in 2 tabanındaki karşılığının bulunması:

$2 \times 0.625 = \mathbf{1} + 0.25$ ← Yüksek anlamlı bit

$2 \times 0.25 \;\;= \mathbf{0} + 0.5$

$2 \times 0.5 \;\;\;= \mathbf{1} + 0$

5/8

$\Rightarrow (0.625)_{10} = (0.101)_2$

$(+22.625)_{10} = (+10110.101)_2 = +1.0110101 \times 2^4$ (Normalize)

Bellekte "Single" olarak:

0	10000011	0110101...000
İşaret +	Yükseltilmiş 127 + 4 üs	Mantis

Mantisin düşük anlamlı bitleri "0" ile dolduruldu. Çünkü solunda nokta (.) var.

Sayının 10 tabanındaki değeri (N):

Single: $N = (-1)^S (1.F) 2^{E-127}$ *Double*: $N = (-1)^S (1.F) 2^{E-1023}$

$N = (-1)^S \left(1 + \frac{F}{2^{23}}\right) 2^{E-127}$ $N = (-1)^S \left(1 + \frac{F}{2^{52}}\right) 2^{E-1023}$

Sıfır için özel bir gösterilim kullanılır: E=0 ve F=0

Örnekler (*Single*):

S	E	F	Anlamı:
0	10010011	10100010000000000000000	$= +1.1010001 \times 2^{10100} = +1.6328125 \times 2^{20}$
1	10010011	10100010000000000000000	$= -1.1010001 \times 2^{10100} = -1.6328125 \times 2^{20}$
0	01101011	10100010000000000000000	$= +1.1010001 \times 2^{-10100} = +1.6328125 \times 2^{-20}$
1	01101011	10100010000000000000000	$= -1.1010001 \times 2^{-10100} = -1.6328125 \times 2^{-20}$

81/128

E (üs) ve F(mantis) özel değerleri:

1. E = 0 ve F = 0 → N = 0 İki tane sıfır var (+,-)
2. E = 255 , F = 0 → N = ±∞
3. E = 255, F ≠ 0 → NaN (*Not a Number*) 0/0 , ∞/∞
4. E = 0, F ≠ 0 → Normalize olmayan sayı

Son durum normalize olarak gösterilemeyen mutlak değeri çok küçük sayılar için kullanılır.

Normalize olmayan sayılar:

Normalize yapı ile mutlak değeri çok küçük olan sayıları göstermek mümkün değildir.

Normalize olarak gösterilebilecek en küçük sayı S =0, E = 0000 0001 , F = 000.... 0
N= $+(1+0)2^{1-127} = 2^{-126}$ 0 ile 2^{-126} arasındaki sayılar normalize olarak gösterilemez.

Normalize olmayan yapı kullanılırsa sayının 10 tabanındaki değeri öncekinden farklı olarak aşağıdaki gibi hesaplanır:

Single: $N = (-1)^S (\frac{F}{2^{23}}) 2^{-126}$ *Double*: $N = (-1)^S (\frac{F}{2^{52}}) 2^{-1022}$

Normalize olmayan en küçük sayı: S =0, E = 0000 0000 , F = 000.... 01
N= $+(1/2^{23})2^{-126} = 2^{-149}$ 0 ile 2^{-149} arasındaki sayılar gösterilemez (*underflow*).

Sınır değerleri:

Mutlak değeri en küçük sayı:

Single: 2^{-149} (Denormalize) , Double: 2^{-1074} (Denormalize)

Mutlak değeri en büyük sayı:

Single: 0 11111110 11111111111111111111111 = $(2-2^{-23})\times 2^{127} \approx 10^{38.53}$ (Normalize)

Double: $(2-2^{-52})\times 2^{1023} \approx 10^{308.3}$ (Normalize)