Floating Point (Kayan Noktalı Sayılar)

BIL-304: Bilgisayar Mimarisi

Dersi veren öğretim üyesi:

Dr. Öğr. Üyesi Fatih Gökçe

Ders kitabına ait sunum dosyalarından adapte edilmiştir: http://csapp.cs.cmu.edu/

Adapted from slides of the textbook: http://csapp.cs.cmu.edu/

Bu slayt setinde: Floating Point

- Arka plan: Kesirli ikili sayılar
- IEEE floating point standardı: Tanımlanması
- Örnekler ve özellikler
- Yuvarlama (Rounding), Çarpma

Floating Point İşlemler: Temel Fikir

- $x +_f y = \text{Yuvarla}(x + y)$

- $x \times_f y = \text{Yuvarla}(x \times y)$

- **Temel fikir**
  - Önce gerçek sonucu hesapla
  - Sonucu istenen hassasiyette verilen bitlere yerleştir.
    - Exponent çok büyükse muhtemelen taşma vardır.
    - Exponent normal ise muhtemelen sonuç frac'e sığacak şekilde yuvarlanır.

Yuvarlama (Rounding)

- **Yuvarlama Modları ($ değerleri yuvarlanarak gösterilmiştir.)**

	$1.40	**$1.60**	**$1.50**	**$2.50**	**–$1.50**
Sıfıra doğru	$1	$1	$1	$2	–$1
Aşağıya doğru (−∞)	$1	$1	$1	$2	–$2
Yukarıya doğru (+∞)	$2	$2	$2	$3	–$1
En yakın çift (varsayılan)	$1	$2	$2	$2	–$2

En yakın çifte yuvarlamaya yakından bakış:

- **Varsayılan yuvarlama modudur**
  - Assembly seviyesine inmeden diğer türleri anlamak zordur.
  - Diğer yuvarlama modları istatistiksel olarak yönelimlidir.
    - Pozitif sayılardan oluşan bir kümenin toplamı tutarlı şekilde yukarı veya aşağı çekilecektir.
- **Diğer onluk basamaklara uygulanışı:**
  - İki muhtemel değerin kesin olarak tam ortasındaysa
    - Yuvarlama en anlamsız dijiti çift yapacak şekilde yapılır
  - Örnek, en yakın yüzde bir yuvarla:

1.2349999	1.23	(Orta noktanın aşağısında)
1.2350001	1.24	(Orta noktadan yukarıda)
1.2350000	1.24	(Tam ortada—yukarı yuvarla)
1.2450000	1.24	(Tam ortada—aşağı yuvarla)

İkili Sayıların Yuvarlanması

- **İkili kesirli sayılar**
  - En anlamsız bit 0 ise “Çift” tir.
  - Yuvarlama pozisyonunun sağ tarafındaki bitler = $100\ldots_2$ şeklindeyse “Tam ortada” dır.

- **Örnekler**
  - En yakın 1/4‘e yuvarla (İkili noktanın 2 bit sağı)

Değer	İkili	Yuvarlanmış	İşlem	Yuvarlanmış Değer
2 3/32	10.00011_2	10.00_2	(<1/2—aşağı)	2
2 3/16	10.00110_2	10.01_2	(>1/2—yukarı)	2 1/4
2 7/8	10.11100_2	11.00_2	(1/2—yukarı)	3
2 5/8	10.10100_2	10.10_2	(1/2—aşağı)	2 1/2

Floating Point (Kayan nokta) Çarpma

- $(-1)^{s1}\, M1\ \ 2^{E1}\ \ \times\ \ (-1)^{s2}\, M2\ \ 2^{E2}$
- **Kesin sonuç:** $(-1)^{s}\, M\ \ 2^{E}$
  - İşaret biti *s*: $s1$ ^ $s2$
  - Anlamlı kısım (Significand) *M*: $M1 \times M2$
  - Üst (Exponent) *E*: $E1 + E2$
- **Düzenlemeler**
  - Eğer $M \geq 2$, *M*'i sağa kaydır, *E'yi* arttır.
  - Eğer *E* aralık dışındaysa, taşma var demektir.
  - *M* 'i frac içerisine yerleşecek şekilde yuvarla
- **Uygulama**
  - En büyük angaryası anlamlı kısımların (significand) çarpılmasıdır.

Floating Point Çarpma Örneği

- **Şu iki sayıyı çarpalım: $1.010_2 \times 2^{-1}$ by $-1.110_2 \times 2^{-2}$**
  - Exp kısmının 8 bit, Frac kısmının 3 bit olduğu bilgisi bize veriliyor.
- **Toplamadan farklı olarak, operandların üst (exponent) lerini topluyoruz.**
  - Sonuç exponent değeri = (–1) + (–2) = –3
- **Bias'lı (yönlendirmeli) gösterimi kullanarak: $E_Z = E_X + E_Y - Bias$**
  - E_X = (–1) + 127 = 126 (8 bit exp için: *Bias = 127*)
  - E_Y = (–2) + 127 = 125
  - E_Z = 126 + 125 – 127 = 124 (değer= –3)
- **Şimdi, anlamlı kısımları çarpalım:**

$$(1.\underbrace{010}_{\text{3-bit fraction}})_2 \times (1.\underbrace{110}_{\text{3-bit fraction}})_2 = (10.\underbrace{001100}_{\text{6-bit fraction}})_2$$

```
      1.010
 ×    1.110
 ----------
       0000
      1010
     1010
    1010
 ----------
   10001100
```

*This slide is adapted from the slides of Dr. M. Mudawar, ICS 233, KFUPM.

Floating Point Çarpma Örneği

- **İşaret bitleri: $S_x \neq S_y$, olduğu için sonucun işaret biti $S_z = 1$ (negatif)**
- **Böylece, $1.010_2 \times 2^{-1} \times -1.110_2 \times 2^{-2} = -10.001100_2 \times 2^{-3}$**
- **Fakat, sonuç: $-10.001100_2 \times 2^{-3}$ normalize halde değildir.**
- **Normalize edelim: $10.001100_2 \times 2^{-3} = 1.0001100_2 \times 2^{-2}$**
  - Sağa 1 bit kaydırıyoruz ve exponenti 1 bit arttırıyoruz.
  - Sağa en fazla 1 bit kaydırılabilir… Niçin?
- **Anlamlı kısmı (significand) en yakına yuvarla:**

  $1.0001100_2 \approx 1.001_2$ (3-bit fraction)

  Sonuç $\approx -1.001_2 \times 2^{-2}$ (normalize edilmiş hali)

  Son durumda: $E_z = 125$ (1111101)

  Sonuç: 1 01111101 001 → 0xBE9

```
  1.000 | 1100
+          1
-------------
  1.001
```

- **Eğer varsa yukarı veya aşağı taşmayı da algılamalıyız!**
  - Burada yukarı veya aşağı taşma yok, çünkü exponent olması gereken aralıkta.

*This slide is adapted from the slides of Dr. M. Mudawar, ICS 233, KFUPM.