

Selamat malam Bapak/Ibu Tim Dosen PSD Fasilkom UI dan teman-teman, mungkin rangkuman yang akan saya sampaikan agak terlambat, mengingat materi floating point merupakan materi minggu 9. Saya harap rangkuman saya dapat bermanfaat untuk Kuis, PR atau Ujian ke depannya.

## 1. Representasi dan Karakteristik Floating Point

Floating point diperlukan untuk merepresentasikan bilangan real. Terdapat dua notasi yang digunakan pada umumnya.

### Fixed-Point

Sama dengan merepresentasikan sebuah bilangan dalam basis dua. Komputer menyediakan N integer bits dan M fraction bits. Pada umumnya 4 integer bits dan 4 fraction bits. Representasi fixed-point ada dua juga, yaitu **Sign-Magnitude** dan **Two's Complement**

#### Contoh Sign-Magnitude:

$$6.75_{10} = 0110.1100_2$$

Akan disimpan sebagai

**01101100<sub>SM</sub>** (Perhatikan bahwa akan dipakai satu bit paling significant pada integer bits untuk menyimpan sign)

$$-7.5_{10} = 0111.1000_2$$

Akan disimpan sebagai

**11111000<sub>SM</sub>** (Perhatikan bahwa sign negatif)

#### Contoh Two's Complement:

$$6.75_{10} = 0110.1100_2$$

Akan disimpan sebagai

**01101100<sub>2S</sub>** (Sama seperti Sign-Magnitude bila positif)

$$7.5_{10} = 0111.1000_2$$

Akan disimpan sebagai

**01111000<sub>2S</sub>**

$$-7.5_{10} = 0111.1000_2$$

Lakukan teknik 2's complement (Flip semua bits dan +1)

**10001000<sub>2S</sub>**

### Floating-Point

Mirip dengan scientific notation.

$$N = \pm M \times B^E$$

Dimana:

M = Mantissa

B = Base

E = Exponent

Dimana Mantissa memiliki titik di sebelah kanan digit paling significant bukan 0.

**Optimisasi Mantissa**, Karena setiap floating point representation akan disimpan dalam binary, diketahui bahwa bit pertama mantissa akan selalu 1.

**Optimisasi Exponent**, Untuk mendapatkan nilai mantissa, N akan digeser geser sehingga titiknya berada di sebelah kanan digit paling significant bukan 0. Gesernya dapat ke kiri atau ke kanan. Sehingga Exponent dapat bernilai negatif. Agar mencakup range bilangan yang luas, maka E harus dapat menyimpan bilangan negatif pula. Akan disimpan dalam bentuk **Biased Exponent** (+127 pada IEEE 754 Single Precision)

## Presisi dan Kasus Khusus

Ada dua jenis presisi yang umum digunakan pada IEEE-754, yaitu single precision dan double precision.

**Single Precision:** Menyimpan 32 bit, 1 sign bit, 8 exponent bits, 23 fraction bits, bias = 127

**Double Precision:** Menyimpan 64 bit, 1 sign bit, 11 exponent bits, 52 fraction bits, bias = 1023

### Contoh Floating-Point Representation (Single Precision):

$$-58.25_{10} = 111010.01_2 = 1.1101001 \times 2^5$$

1 Sign bits = 1 (**Negative**)

8 Biased Exponent bits =  $(127+5) = 132 = 10000100_2$

23 Fraction bits = 110 1001 0000 0000 0000 0000 (**Perhatikan bahwa bit pertama sebelum titik diabaikan**)

1 100 0010 0 110 1001 0000 0000 0000 0000

Hexadecimal = 0xC269000

Ada pula kasus lain selain bentuk biasa:

- 0, Sign = X, Exponent = 00000000, Fraction 000000000000000000000000
- $\infty$ , Sign = 0, Exponent = 11111111, Fraction 000000000000000000000000
- $-\infty$ , Sign = 1, Exponent = 11111111, Fraction 000000000000000000000000
- NaN, Sign = X, Exponent = 11111111, Fraction non-zero

Ada pula kasus denormalized, yaitu saat bilangan yang akan direpresentasikan terlalu kecil (Terlalu mendekati 0) sehingga exponent tidak sanggup mencapainya, Dipastikan dalam representasi Floating pointnya pada single precision menjadi  $N = \pm M \times B^{-126}$

Perhatikan, bahwa Mantissa kini most significant bitsnya tidak 1 lagi. Karena kasus tersebut, bisa saja terjadi underflow, atau bilangan terlalu kecil untuk direpresentasikan.

## Rounding

Ada empat jenis rounding floating point yang umum dipakai:

- Down: Pembulatan selalu ke 0
- Up: Pembulatan selalu ke 1
- Toward Zero: Pembulatan ke 0 terdekat (Sama seperti aturan baku scientific notation)
- To nearest: Pembulatan ke desimal terdekat

## 2. Operasi Floating Point

Langkah-langkah dalam melakukan operasi floating point ialah

1. Cari exponent bits dan fraction bits
2. Jadikan bentuk basis 2
3. Bandingkan pangkat
4. Geser mantissa sesuai kebutuhan agar pangkatnya sama
5. Jumlahkan Mantissa
6. Normalisasikan mantissa

7. Pembulatan (Khusus saat melebihi kapasitas fraction bits)

8. Jadikan format floating point lagi.

Sekian rangkuman singkat saya, saya akan berusaha membuat rangkuman untuk minggu 10 juga, mohon maaf apabila ada kesalahan. Saya harap rangkuman ini dapat bermanfaat bagi teman-teman. Terima kasih.

Salam,

Hocky