浮動小数点数（実数）の表現

コンピュータシステム 第３回講義

http://www.cuc.ac.jp/~miyata/classes/system/

本日の内容

- 少数点のある2進数を10進数に変換
- 少数点のある10進数を2進数に変換
- 浮動小数点数
- 浮動小数点数の内部形式
- 浮動小数点数と誤差

小数がある２進数⇒１０進数

- 小数点のある2進数 $b_n \cdots b_1 b_0 \,.\, a_1 a_2 a_3 \cdots$ は
  $2^n b_n + \cdots + 2^1 b_1 + 2^0 b_0 + 2^{-1} a_1 + 2^{-2} a_2 + 2^{-3} a_3 + \cdots$
  を表している

- 例：2進数 １０１．０１１ は

$$1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3}$$
$$= 1 \times 4 + 0 \times 2 + 1 \times 1 + 0 \times \frac{1}{2} + 1 \times \frac{1}{4} + 1 \times \frac{1}{8}$$
$$= 4 + 1 + 0.25 + 0.125$$
$$= 5.375$$

- 練習：2進数 １１．１０１１ を10進数にしなさい

小数がある１０進数⇒２進数

- 小数点のある１０進数から２進数への変換方法
  - 整数部と小数部に分ける（整数部はふつうに2進数に変換）
  - 小数部に２を掛けて整数部（０か１）を取り出す
  - 小数部が０になるか，適当な精度まで求めたら終了。
- 例：１０進数の「３．７」は

  ３→「１１」

  ０．７ × ２ → **1**．４
  ０．４ × ２ → **0**．８
  ０．８ × ２ → **1**．６
  ０．６ × ２ → **1**．２
  ０．２ × ２ → **0**．４
  以下，０１１０を繰り返す

  よって，１１．１０１１００１１００１１００１１０・・・
- 練習：10進数の「1.1」を2進数で表せ。ただし小数点以下１１桁目以降を切り捨てて，小数点以下10桁まで求めなさい。

浮動小数点数（floating point number）

- 限られた情報量（ビット数）で広い範囲の数値を表現する方法

$$\pm m \times R^e$$

の形で実数を表現する。mを仮数部 (mantissa)， eを指数部 (exponent) と呼ぶ。また Rは基数 (radix)と呼ばれ，コンピュータでは $R = 2$ が使われる（人間は一般的に $R = 10$ を使う）。

例：基数R＝１０とする

１２３＝１２.３×１０^{1}＝１.２３×１０^{2}＝０.１２３×１０^{3}

- ０以外の数は指数を調節して$1 \leq m < R$ となるように表現することができる。これを正規化表現という。

  上の例では 「１.２３×１０^{2}」が正規化表現

浮動小数点数の例（１０進数）

- 簡単のため基数１０（１０進数）で説明する。
- 仮数部3桁，指数部を符号付で1桁（－９～９）として正規化表現だけを許すものとする。
  - 光の速さ（m/s）：

    ２９９７９２４５８ → ３.００×１０^{8}　　※四捨五入した
  - ロト６の１等当選確率

    ０.０００００００１６４０２９７７ → １.６４×１０$^{-7}$
- 10進4桁（＋符号情報2bit）の情報量で幅広い数を扱うことができる
  - 最大 ９.９９×１０^{9} = 9,990,000,000
  - 正の数で最小１.００×１０$^{-9}$ = 0.000000001
  - 負の数で最大－１.００×１０$^{-9}$ = -0.000000001
  - 最小 -９.９９×１０^{9} = -9,990,000,000

浮動小数点数どうしの演算

- 簡単のため仮数部2桁とする
- 乗算（仮数部どうしを乗算，指数部は加算）
  - ８.４×１０^{5} × ２.１×１０^{2}

    （８.４×２.１）×１０$^{5+2}$

    ＝１７.６４×１０^{7}

    → １.８×１０^{8}
- 除算 （仮数部どうしを除算して指数部は減算）
- 加算（指数部をそろえてから計算）
  - ８.４×１０^{5} ＋ ２.１×１０^{2}

    ８.４×１０^{5} ＋ ０.００２１×１０^{5}

    ＝８.４０２１×１０^{5}

    → ８.４×１０^{5}
- 減算（指数部をそろえてから計算）

浮動小数点数の例（２進数）

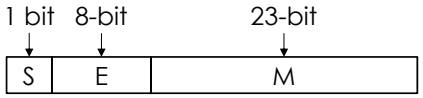
- 次の2進数を正規化し仮数部４桁の浮動所数点数にせよ。ただし仮数部の小数点以下第４桁は０捨１入せよ。
  - $(１１０１００１０)_2$

    → $(１.１０１)_2 \times ２^{7}$
  - $(０.００１１００１１００)_2$

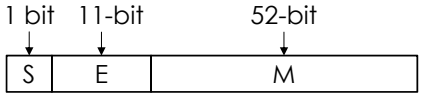
    → $(１.１０１)_2 \times ２^{-3}$
  - ※ 正規化すると先頭は“１.”で始まることに注意（ただし０は例外！）
- 練習：次の2進数を正規化し仮数部4桁の浮動小数点数にせよ。ただし仮数部の小数点以下第４桁は０捨１入せよ。
  - $(１１１０１００)_2$
  - $(０.０００１０１０１)_2$

浮動小数点数の内部形式

- IEEE754 単精度浮動小数点（３２bit）

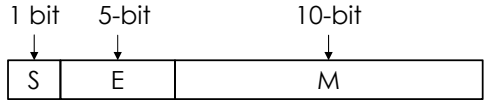


  - S：仮数部の符号（０：正，１：負）
  - E：指数部
  - M：仮数部（の少数部）
  - 正規数として表現できる範囲：およそ$\pm 10^{-38} \sim \pm 10^{38}$
  - 精度：10進数で7桁程度
- IEEE754 倍精度浮動小数点（６４bit）



  - 正規数として表現できる範囲：およそ$\pm 10^{-308} \sim \pm 10^{308}$
  - 精度：10進数で16桁程度

IEEE754 半精度浮動小数点数(16bit)



$(-1)^S \times (1+M) \times 2^{E-15}$　（正規化数）

$(-1)^S \times (0+M) \times 2^{-14}$　（非正規化数）

- S：仮数部の符号（０：正，１：負）
- E：指数部
  - 符号なし整数（0≦E≦31），オフセット表現（オフセット15）
  - E=0 のときは０または非正規化数，E＝255のときは，±無限大または非数（NaN）
- M：仮数の少数部
  - E=0のとき，正規化されていない仮数の少数部
  - 1≦E≦30のとき，正規化された仮数の少数部
  - E＝31のとき，無限大（M=0），NaN（M≠０）
- 特殊なハードウェアでしか使われていない（あまり一般的ではないが，試験問題向きなので取り上げた）

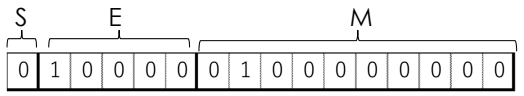
補足

- 2進数の場合（０以外の数は）正規化すると必ず”１.“で始まるので仮数部には小数部（”１.XXXXX“のXXXXXの部分）だけを格納する。
  - ０は例外扱い。＋０と－０がある。指数部０の場合非正規数。
  - 無限大や非数（NaN）がある
- 単精度浮動小数点数の仮数部は実質的に１１ビットの精度
  - 10進数で３桁程度の精度

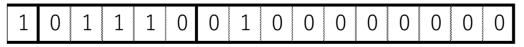
$$\log_{10} 2^{11} = 11 \log_{10} 2 = 11 \times 0.301 = 3.311$$

- 正規化数として表現できる正の数の範囲は $6 \times 10^{-5} \sim 6 \times 10^{4}$ 程度
- 指数部を２の補数表現にせずに，オフセット表現（１５足した値）にするのは，数値の大小の比較を簡単にするため。

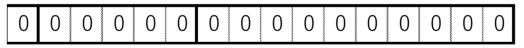
半精度浮動小数点数の例



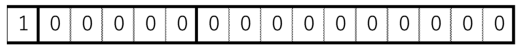
$(1.01)_2 \times 2^{16-15} = (10.1)_2 = 2.5$



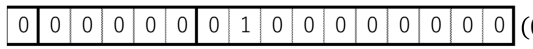
$(-1.01)_2 \times 2^{14-15} = (-0.101)_2 = -0.625$



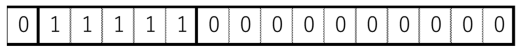
$+0$



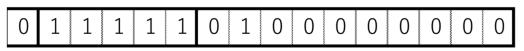
-0



$(0.01)_2 \times 2^{-14} = 2^{-16}$　（非正規数）



$+\infty$



NaN（非数）

例題

- 「-10.75」 を IEEE754半精度浮動小数点形式で表せ

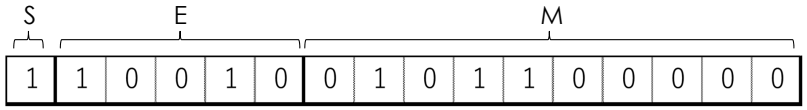
  M: 10.75を2進数に変換して正規化

  $(10.75)_{10} = (1010.11)_2 = (1.01011)_2 \times 2^3$

  E： 指数の3にオフセット１５を足すと１８

  $(18)_{10} = (10010)_2$

  S: 符号部は１



浮動小数点数と誤差（１）

- 丸め誤差
  - 仮数部の桁が有限なので，仮数部に入りきらない桁は四捨五入や切り上げ，切り捨てされることになる。このときに生じる誤差。
- オーバーフロー（overflow, 桁あふれ）
  - 計算結果（の絶対値）が大きすぎて，正規数として表現できる範囲を超えること
- アンダーフロー（underflow)
  - 計算結果（の絶対値）が小さすぎて，正規数として表現できる範囲を超えること

浮動小数点数と誤差（２）

- 情報落ち誤差
  - （絶対値の）大きな数と小さな数の加減算を行ったときに，小さいほうの値が演算結果に反映されないことによって生じる誤差。
  - スライド７参照
  - データの総和を出すような場合，小さいものから足すことで回避できる場合もある
- 桁落ち
  - （絶対値の）近い２数で加減算を行ったときに精度が失われること

```
   1．１１１０１１０１０１
ー）1．１１１０１１０００１
   0．０００００００１００
```

１１ビットの情報が3ビットに減少！