

特殊の数

- ± 0 : $M = 0$, $E = -1023$, $S = 0 (+0) \text{ or } 1 (-0)$
仮数ビットはすべて0 指数ビットはすべて0
- $\pm \infty$: $M = 0$, $E = 102$, $S = 0 (+\infty) \text{ or } 1 (-\infty)$
↑
指数ビットはすべて1 符号は無関係
- NaN: $M \neq 0$, $E = 102$, $S \in \{0, 1\}$
↑
指数ビットはすべて1 ↓

⑧ 浮動小数点計算における誤差

① 丸め誤差 (rounding error)

(例) $\pi \rightarrow \boxed{3.1416} \times 10^0$
↑
仮数部: 十進5桁 計算機内部に 16-bit (guard digit/bit) をつけて丸めを行う。

② 相消し誤差 (cancellation error)

(例) 仮数部 十進9桁

1.23456789	$\times 10^0$
- 1.23450000	$\times 10^0$
0.00006789 $\times 10^0$	

↓ 正規化

有効桁数が9桁から → 6.78900000 $\times 10^{-5}$
 4桁へ減少。 これらの桁は正確でない。

★ 絶対値の大きさが近い2数の加減算で生じることがある。

③ 情報落ち

(例) 仮数部 十進3桁

$$\begin{cases} x = 1.00 \times 10^4 \\ y = -1.00 \times 10^4 \\ z = -1.00 \times 10^0 (= 1.00) \end{cases}$$

271. $x + y + z$ を計算する。

$$(1) \quad \frac{x + (y + z)}{\quad}$$

加減算の際は
揃えろええ。

$$\textcircled{1} \quad \begin{array}{r} y = -1.00 \times 10^4 \\ + \quad z = -1.00 \times 10^0 \end{array}$$

$\Rightarrow$

$$\begin{array}{r} -1.00 \quad \times 10^4 \\ + \quad -0.0001 \times 10^4 \\ \hline -1.0001 \times 10^4 \end{array}$$

3桁に丸める

$\Downarrow$

$$y + z = -1.00 \times 10^4$$

$$\textcircled{2} \quad \begin{array}{r} y + z = -1.00 \times 10^4 \\ + \quad x = 1.00 \times 10^4 \\ \hline 0.00 \times 10^4 \end{array}$$

$\rightarrow$ 答え: 0.0

$$(2) \quad \frac{(x + y) + z}{\quad}$$

$$\textcircled{1} \quad \begin{array}{r} x = 1.00 \times 10^4 \\ + \quad y = -1.00 \times 10^4 \\ \hline x + y = 0.00 \times 10^4 = 0.00 \times 10^0 \end{array}$$

$$\textcircled{2} \quad \begin{array}{r} x + y = 0.00 \times 10^0 \\ + \quad z = -1.00 \times 10^0 \\ \hline -1.00 \times 10^0 \end{array} \rightarrow \text{答え: } \boxed{-1.0}$$

計算の
順序で
符号が
異なる!

★ 桁の加減算の場合、なるべく有効桁が近い数同士を
とめて計算することが大切。

2.1.3. グレイコード $\Rightarrow$ 省略

(「九連環」を試したい人の坂井先生へ!)

2.1.4. -2 進表記 $\Rightarrow$ 省略

2.1.5 多精度整数の表記と加算

⑩ 今後、整数や有理数などの演算は正確に行うことと
前 提とする。

- ・ 1ワードに収まり切れない数を持つ場合。
→ 複数ワードを用いて1つの数表現し、演算を行う
これを (多精度演算) という。
(多倍長演算)

⑪ 非負整数の多倍長演算 (多倍長整数)
(加算) 精度

{ データの表現
演算のアルゴリズム } 考える必要がある。

・ データの表現

- ① 数値を表すワード数 $\rightarrow k$
- ② 数値を表すワード $\rightarrow a_0, a_1, \dots, a_{k-1}$

s.t.

$$v = \sum_{i=0}^{k-1} a_i \cdot 2^{ni} \quad | n: \text{ワードの大きさ (bit)}$$

$$\begin{array}{ccccccc} a_{k-1} & a_{k-2} & \dots & a_1 & a_0 \\ \boxed{\text{1111}} & \boxed{\text{1111}} & \dots & \boxed{\text{1111}} & \boxed{\text{1111}} \end{array}$$

↓
m桁の2進数を並べたものと思う。
[$a_0 \dots a_{k-1}$] と表した。

- ・ 2進数のデータを、実際にどのような形でメモリに保持するかについて、今回は議論しない。
このような議論は データ構造 に関する場で取り扱われる。

・ $a \in \mathbb{N}$ ($a \geq 0$) を2進表記するのに何ワード必要か?

・ a を2進表記したときのビット長 $\rightarrow \lceil \lg(a+1) \rceil$

- 1ワードが n ビットの時. a を 2進表記するのに必要なワード長 $\rightarrow \left\lceil \frac{\lg(a+1)}{n} \right\rceil =: \text{len}(a)$
 a の長さ

- + $\text{len}(a)$ そのものを格納するワード $\rightarrow 1$.

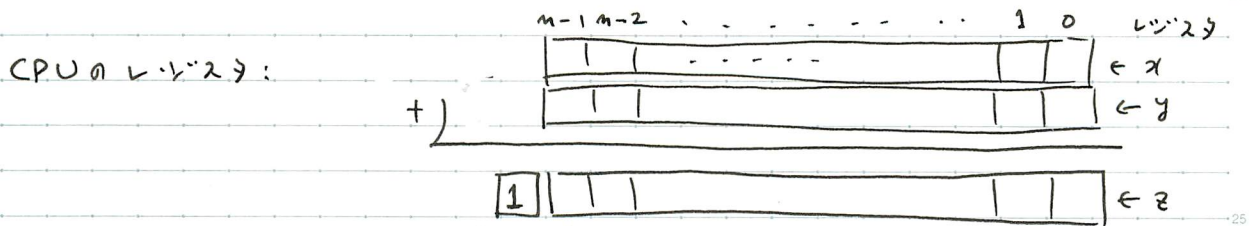
- 計. $1 + \text{len}(a)$ ワード を用いる.

- 問題 2.4 : $a, b \in \mathbb{N}$ $n \geq 1$. $a+b$ と ab のビット数の見積り $\rightarrow$ 計算結果の格納に用いる.

⊗ 多倍長整数の加算

- 用いる道具: 1ワードの整数 (2進数) 2つの加算.

- 桁あふれ (繰り上がり) の検出 $\rightarrow$ 補助演算装置.



桁あふれ (繰り上がり) の有無を $\gamma \in \{0, 1\}$ とし、
 他を補助演算装置に格納.

その下位 n ビットを z に格納. (桁)

$$\begin{aligned}
 \text{2進数,} & \quad x + y = \gamma \cdot 2^n + z. \\
 \text{2進数} & \quad [z, \gamma] \leftarrow x + y \\
 \text{を表す.} &
 \end{aligned}$$

表す

- k ワードの多倍長整数 a と b の加算.