計算機構成論 第12回
―算術演算の実行(3)―

大連理工大学・立命館大学 国際情報ソフトウェア学部

大森 隆行

講義内容

- 算術演算の実行
  - 小数の2進数表記
  - 浮動小数点数の加算
  - (浮動小数点数の乗算)
  - 誤差と丸め

2進数の小数　復習

- 10進数と同様に小数点を用いる
- 小数第1位が2^{-1}に相当

- 例) $11.11_2 = 1*2^1+1*2^0+1*2^{-1}+1*2^{-2}$
  $= 2+1+0.5+0.25$
  $= 3.75_{10}$
  $0.0011_2 = 1*2^{-3}+1*2^{-4}$
  $= 0.125+0.0625$
  $= 0.1875_{10}$

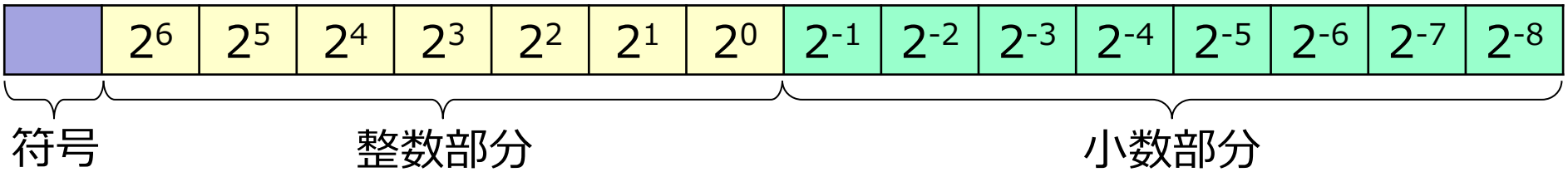
2進数の小数の問題点　復習

- 誤差の発生
  - 例) $0.1_{10} = 0.0001100110011001100...{}_{2}$
    $= 0.0625 + 0.03125 + ...$

10進数の有限小数を有限桁で
表すことができない！

小数の2進数表記

- 固定小数点形式
  - 整数部分と小数部分のビット数を予め固定



  - 利点：わかりやすい
  - 欠点：表現可能な値の範囲が狭い
    - 上の例での表現可能な小数部分の最小値(非零): 2^{-8}
    - 上の例での最大の絶対値: 2^7-2^{-8} (=127.99609375)

小数の2進数表記

- 科学記数法 (scientific notation)
  - 整数部分は1桁のみにする
  - 仮数 × 基数^{指数} という形式で表現

(例) 3.1415_{10} → $3.1415_{10}\times10^{0}$

0.01234_{10} → $1.234_{10}\times10^{-2}$

3155760000_{10} → $3.15576_{10}\times10^{9}$

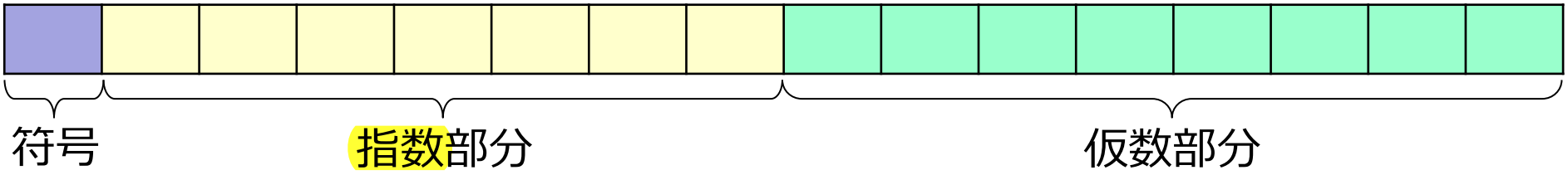
正規化

2進数でも、

$1.010101_2 \times 2^{-2}$ のように表記

小数の2進数表記

- 浮動小数点 (floating point) 形式

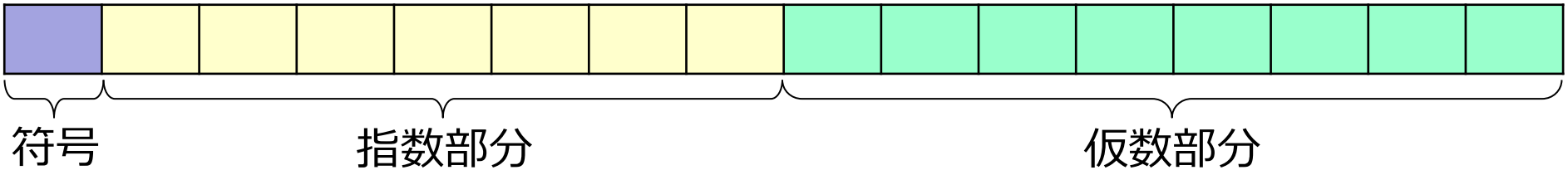


小数点の位置が流動的

- 表現可能な数値の幅が広い
- 表現できる数の刻み幅が流動的
  - e.g., 1.0×2^{-64}, 1.0×2^{63}

小数の2進数表記

- 浮動小数点 (floating point) 形式



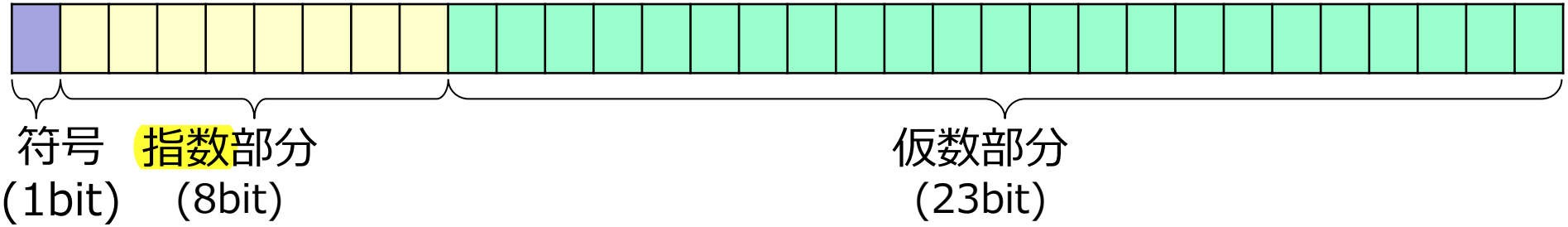
- 表現可能な数値の幅が広い

But

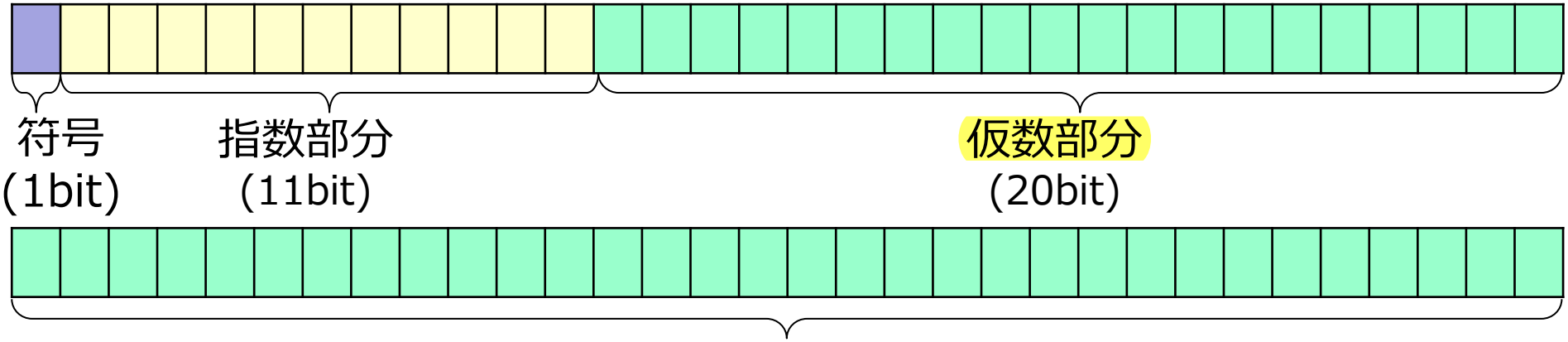
- オーバーフローは発生する
  - 数値の指数が指数部分に収まりきらなくなること
- アンダーフローも発生する
  - 負の指数が大きすぎて指数部分に収まりきらなくなる

浮動小数点形式の数値表現

- IEEE754による規定 **$(-1)^{符号} \times (1+仮数) \times 2^{(指数-ゲタ)}$**
  - 単精度 (32bit)　C言語のfloat型に相当



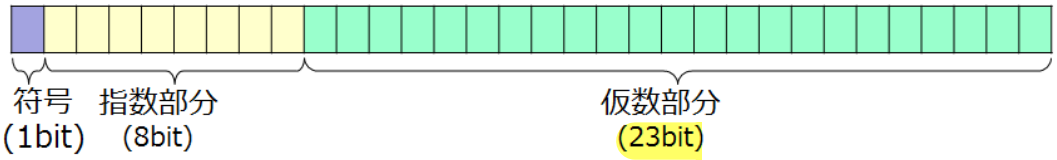
  - 倍精度 (64bit)　C言語のdouble型に相当



浮動小数点形式の数値表現

- 単精度

$$(-1)^{符号} \times (1+仮数) \times 2^{(指数-ゲタ)}$$



- 正規化後、仮数(2進数)の先頭は必ず1
  →仮数部分23bitの中では持たない
  - 仮数部は0～1の小数を示す
- 指数ビットで大小比較ができない
  - -1:11111111, 1:00000001
    → 最小値が00000001になるように
    ゲタ(bias)をはかせる
  - 単精度の場合、実際の値 +127 の値を指数部で保持

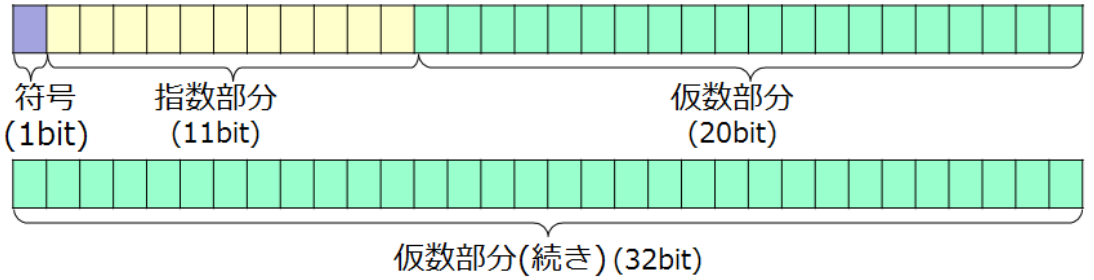
下駄(げた):伝統的な木製の履物

単精度の 最小値は、$1.00..00_2 \times 2^{-126}$ 、最大値は、$1.11..11_2 \times 2^{127}$

浮動小数点形式の数値表現

倍精度

$$(-1)^{符号} \times (1+仮数) \times 2^{(指数-ゲタ)}$$



- 正規化後、仮数(2進数)の先頭は必ず1
  - 仮数部は0～1の小数を示す
- 指数部分にはゲタ(1023)を履かせる

倍精度の 最小値は、$1.00..00_2 \times 2^{-1022}$ 、最大値は、$1.11..11_2 \times 2^{1023}$

浮動小数点形式の数値表現

特別な値の表現

指数	仮数	内容
全て0	0	0
全て0	≠0	±不正規化数
1-254	任意	±浮動小数点数
全て1	0	±∞
全て1	≠0	非数 (NaN)

※不正規化数：0に近い値を表現

確認問題

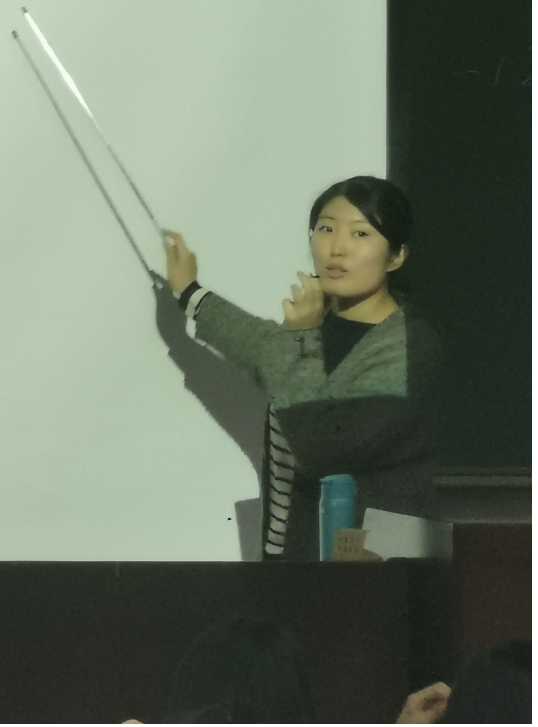
	符号	指数部分	仮数部分
単精度	1bit	8bit	23bit
倍精度	1bit	11bit	52bit

- (1) -0.75_{10}を単精度で表現せよ。
- (2) 0.75_{10}を倍精度で表現せよ。
- (3) 1.75_{10}を単精度で表現せよ。
- (4) $-\infty$を倍精度で表現せよ。
- (5) 単精度の場合のNaNの指数部を答えよ。

確認問題

	符号	指数部分	仮数部分
単精度	1bit	8bit	23bit
倍精度	1bit	11bit	52bit

- (1) -0.75_{10}を単精度で表現せよ。
  - $-0.75_{10}=-0.11_2=-1.1_2\times2^{-1}$ → 指数部は-1+127
  - 1011 1111 0100 0000 0000 0000 0000 0000
- (2) 0.75_{10}を倍精度で表現せよ。
  - 指数部は-1+1023
  - 0011 1111 1110 1000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
- (3) 1.75_{10}を単精度で表現せよ。
  - $1.75_{10}=1.11_2\times2^0$ → 指数部は0+127
  - 0011 1111 1110 0000 0000 0000 0000 0000
- (4) $-\infty$を倍精度で表現せよ。
  - 1111 1111 1111 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
- (5) 単精度の場合のNaNの指数部を答えよ。
  - 1111 1111



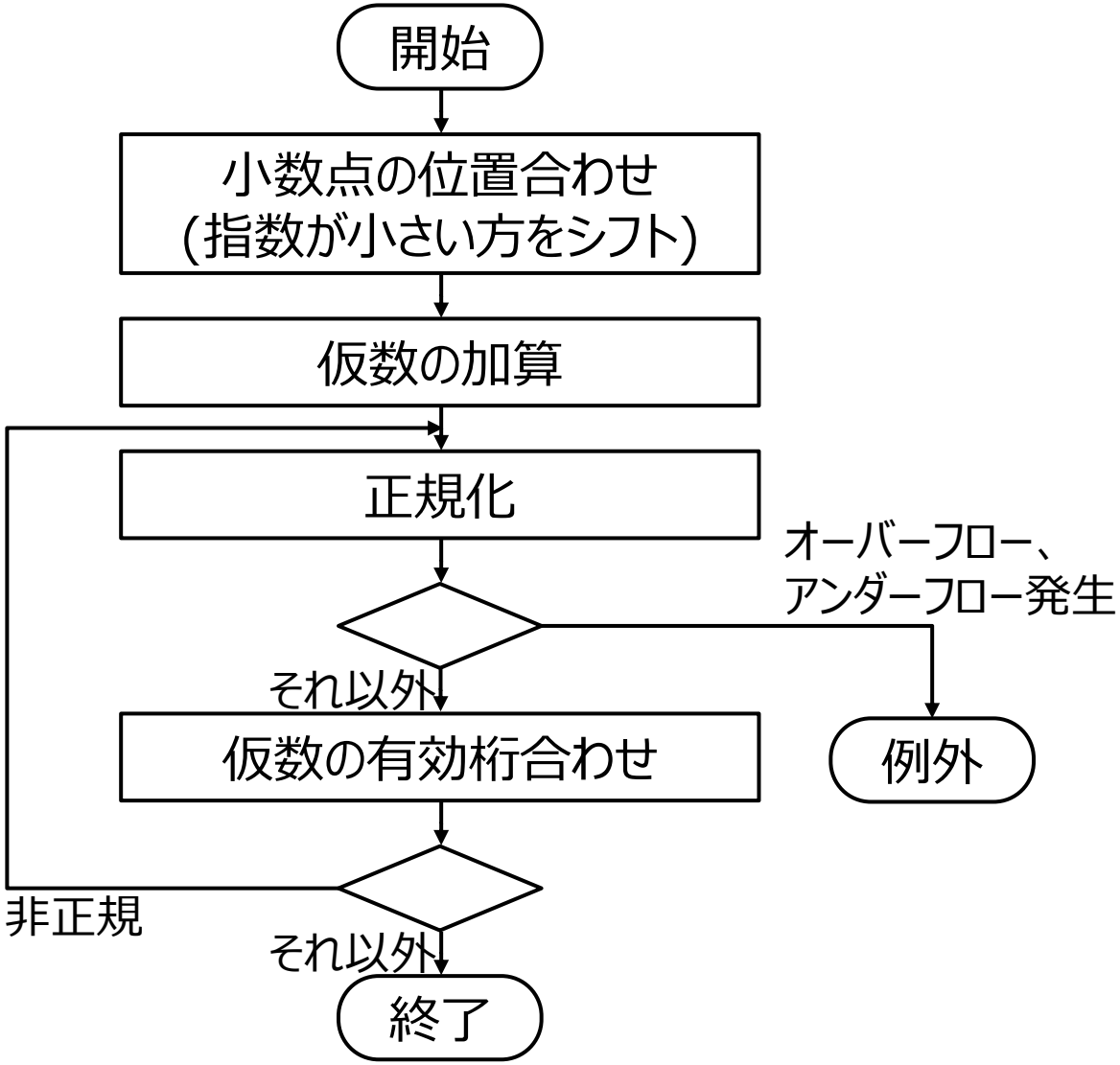
確認問題

単精度・倍精度の符号・指数部分・仮数部分の
ビット幅は覚えなくても良いですが、
ゲタは自分で導出できるようになっておいてください
$(2^{n-1}-1)$ (n:指数部のビット数)

講義内容

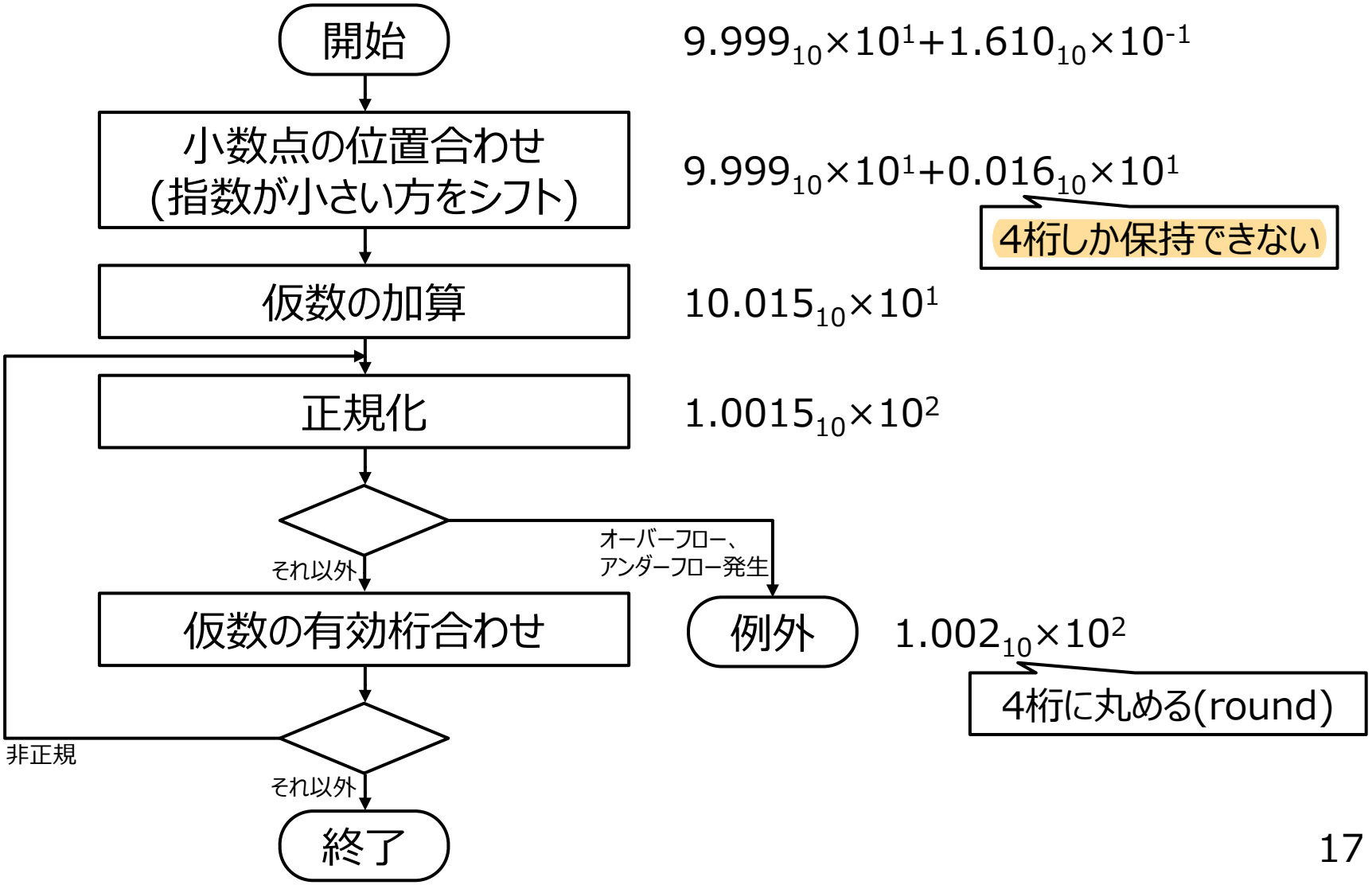
- 算術演算の実行
  - 小数の2進数表記
  - 浮動小数点数の加算
  - (浮動小数点数の乗算)
  - 誤差と丸め

浮動小数点の加算

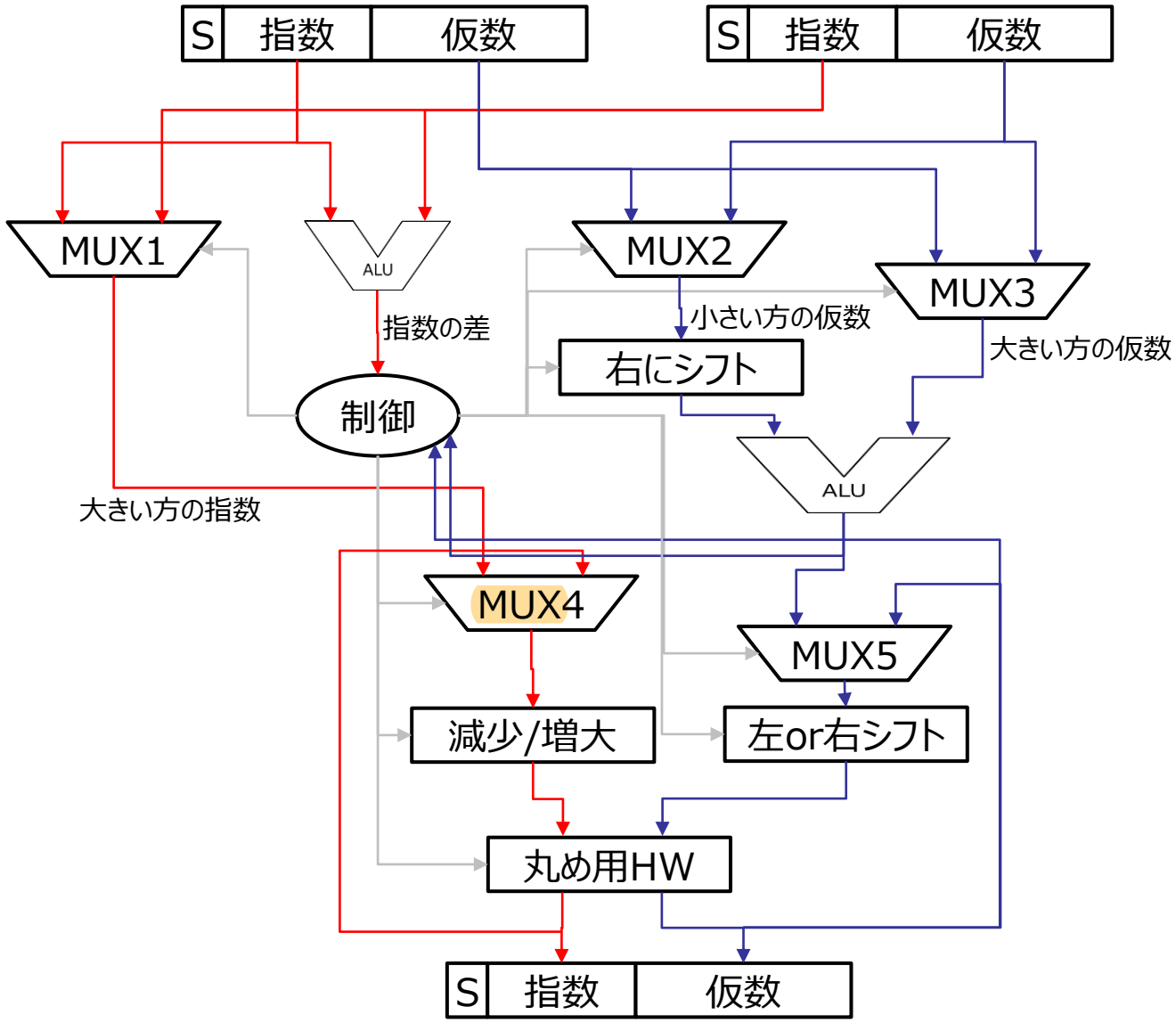

開始
小数点の位置合わせ
(指数が小さい方をシフト)
仮数の加算
正規化
オーバーフロー、
アンダーフロー発生
それ以外
仮数の有効桁合わせ
例外
非正規
それ以外
終了

浮動小数点の加算

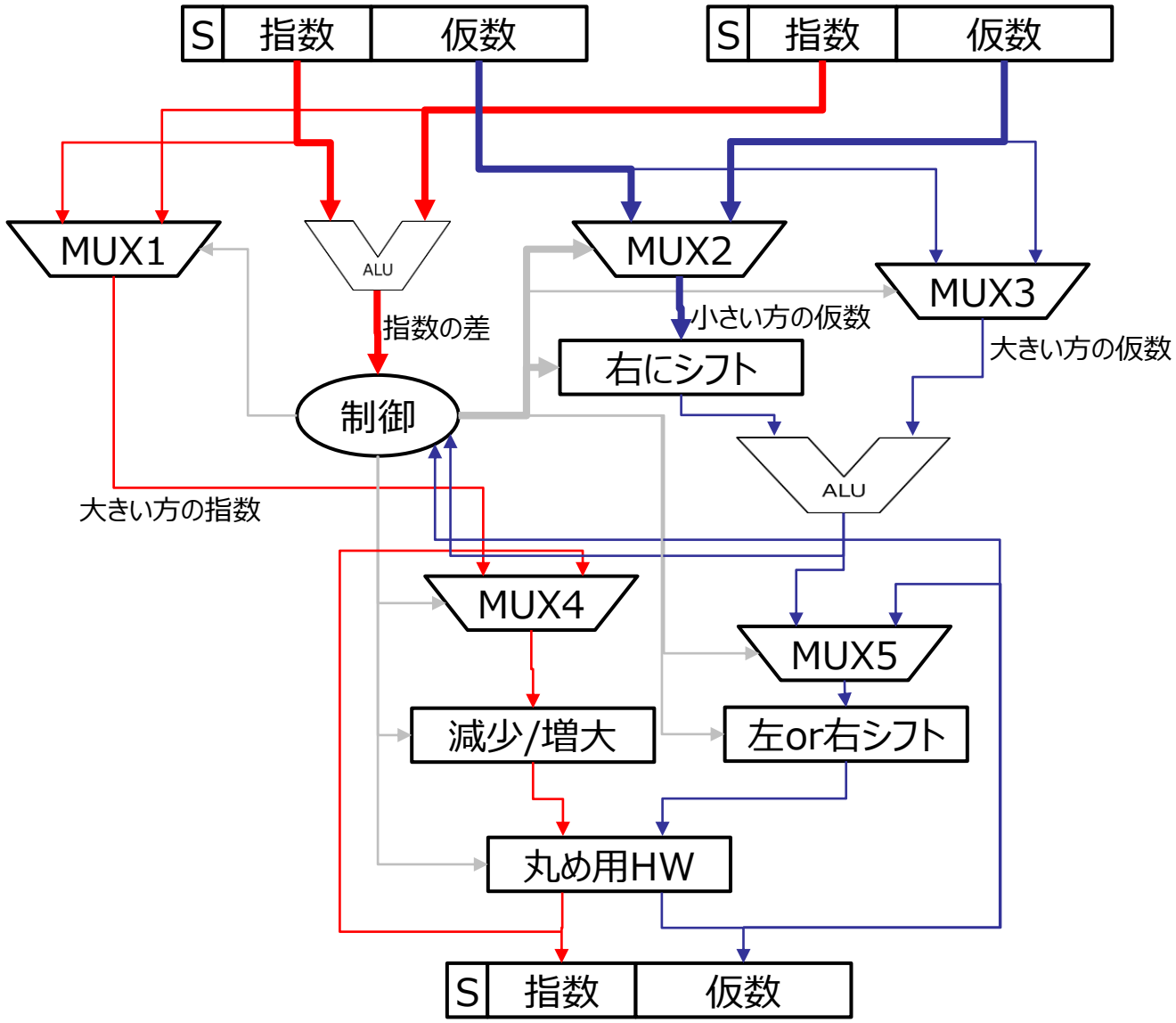
簡単のため、10進数で説明。
仮数4桁、指数2桁と仮定。



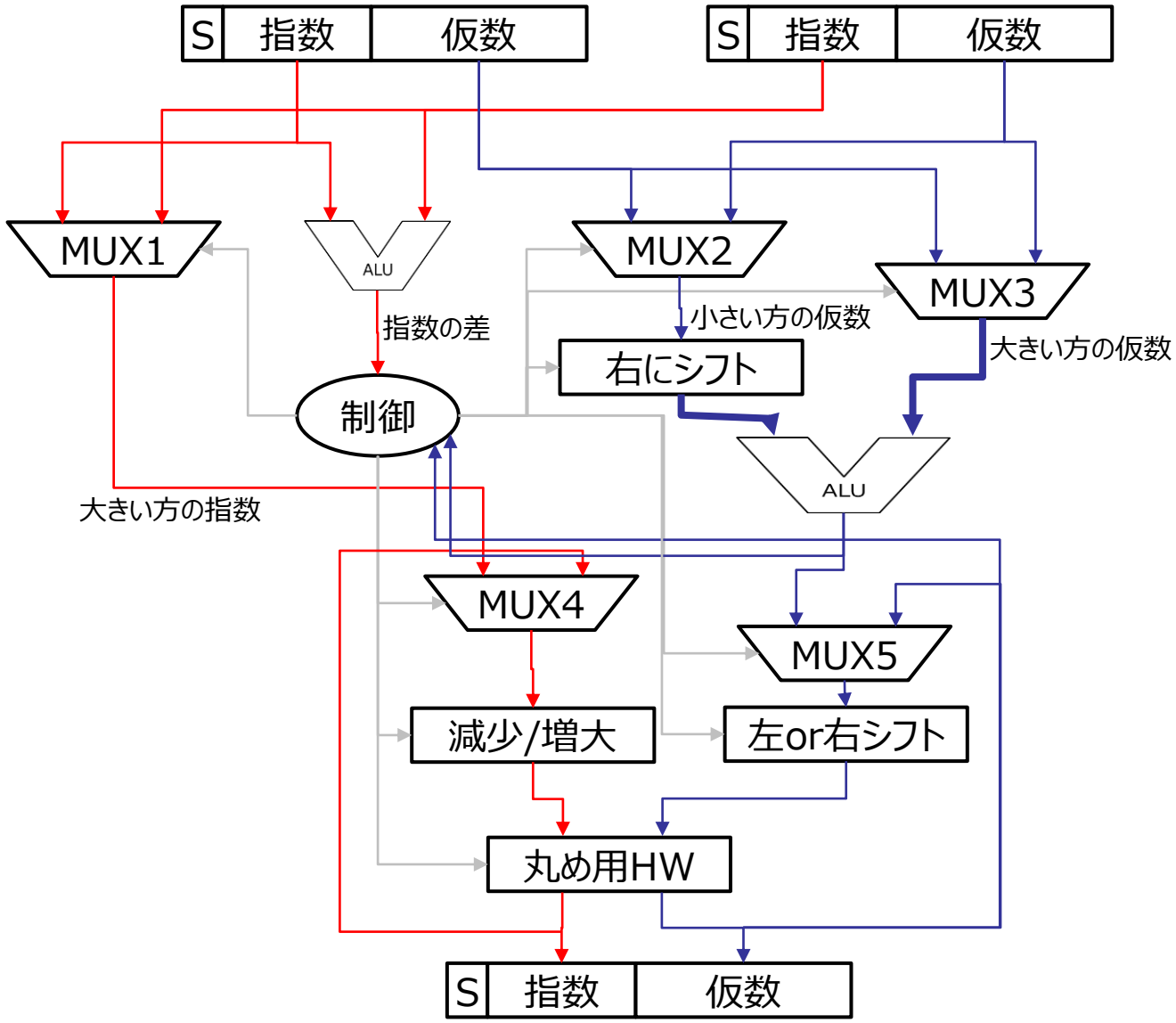
浮動小数点の加算ユニット



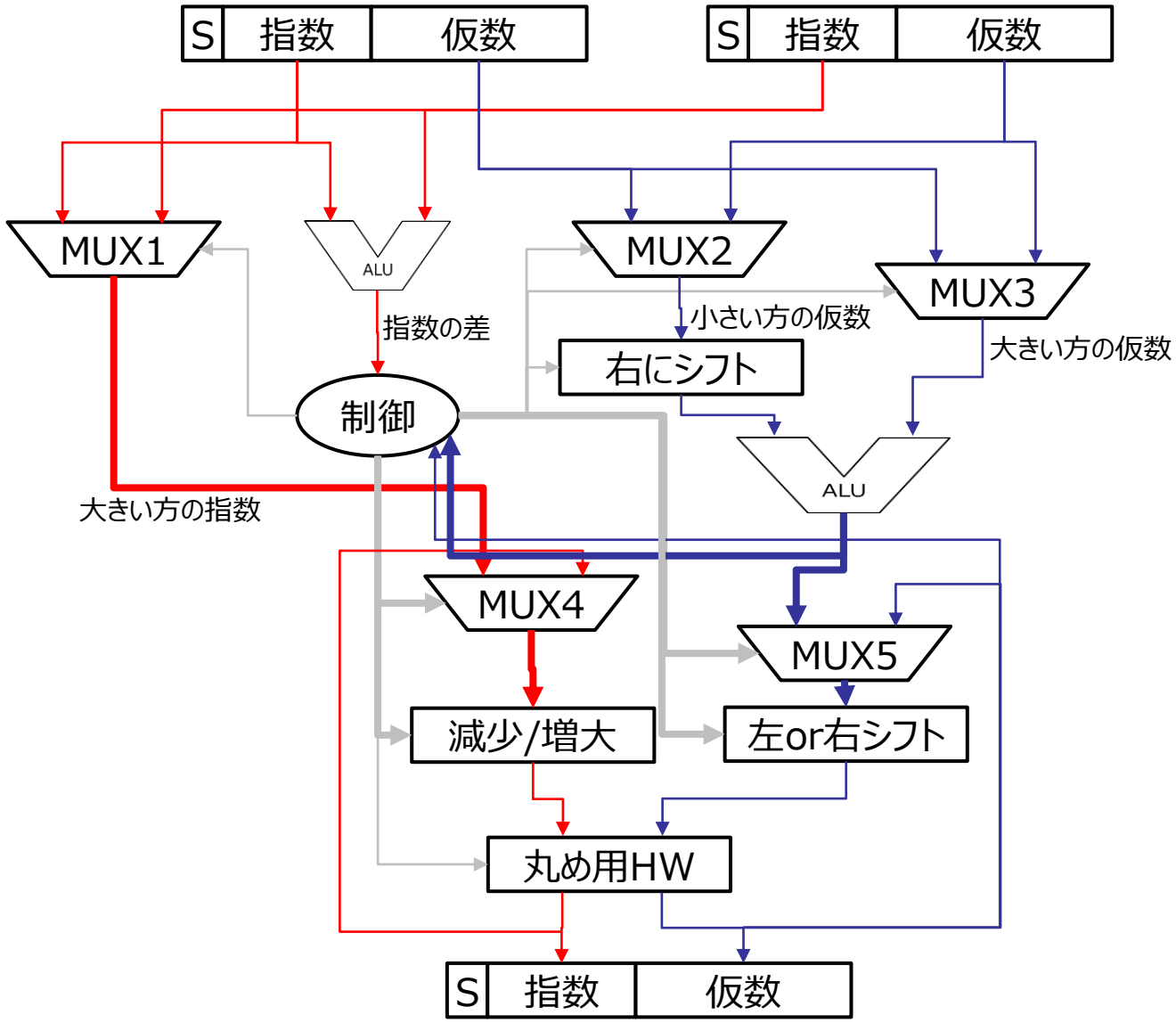
浮動小数点の加算ユニット



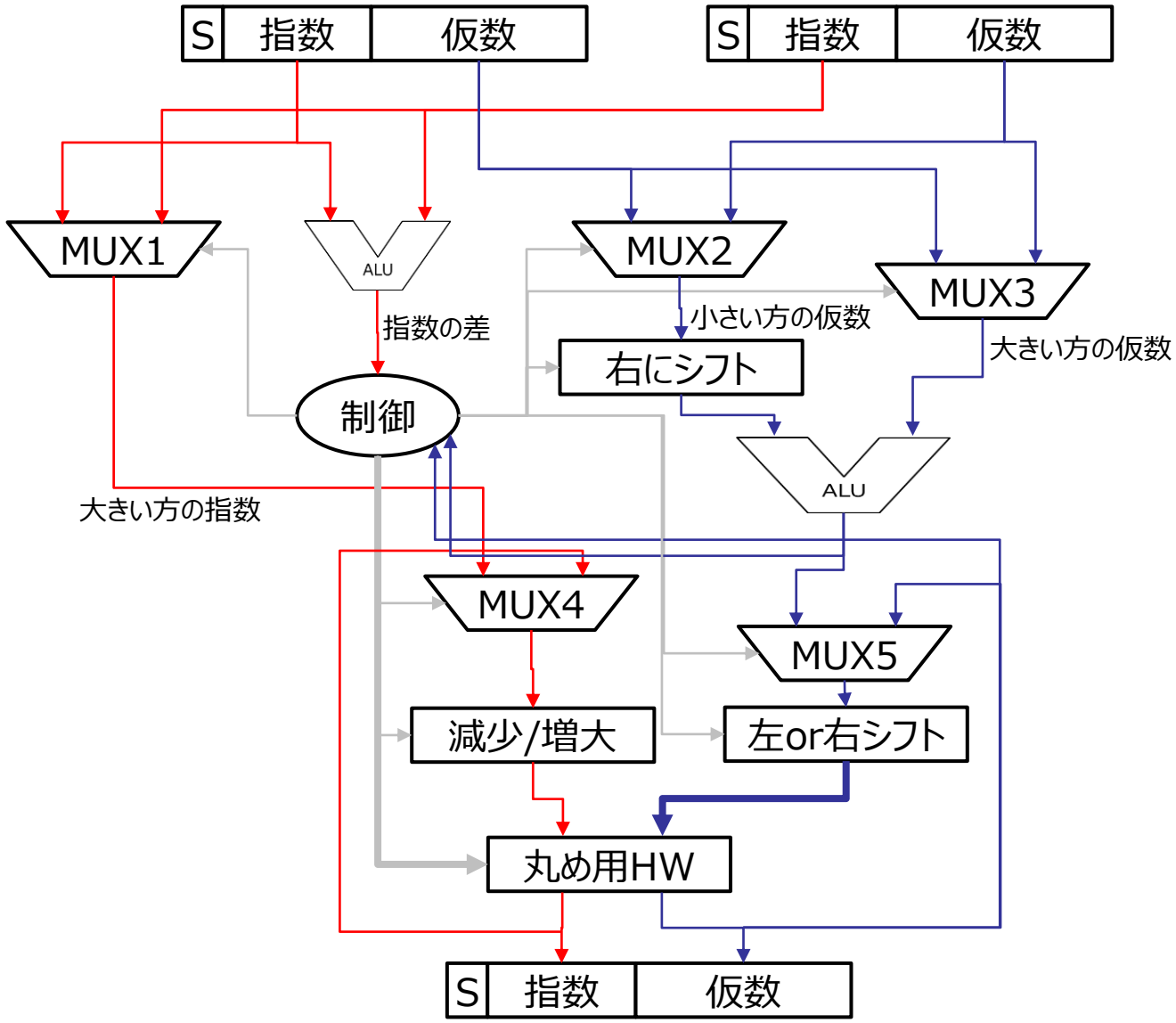
浮動小数点の加算ユニット



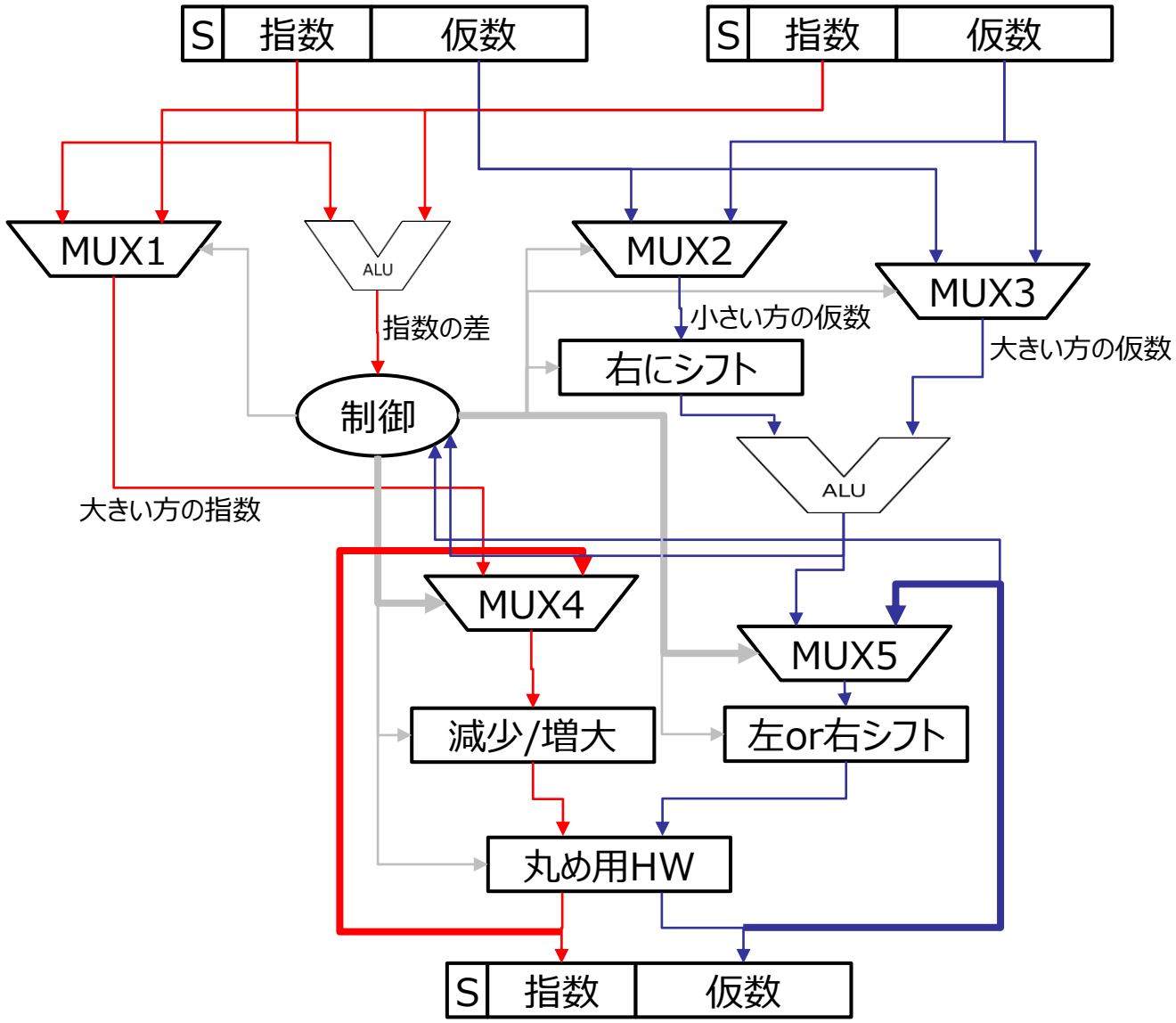
浮動小数点の加算ユニット



浮動小数点の加算ユニット



浮動小数点の加算ユニット



講義内容

- 算術演算の実行
  - 小数の2進数表記
  - 浮動小数点数の加算
  - (浮動小数点数の乗算)
  - 誤差と丸め

浮動小数点の乗算 (発展)

- 人間の乗算と同様に行う
  - 指数同士を加算(ゲタに注意)
  - 仮数の計算
  - 正規化
  - 丸め
  - 符号の決定

  - 詳細は教科書で確認を
    (興味があれば図書館で確認してください)

講義内容

- 算術演算の実行
  - 小数の2進数表記
  - 浮動小数点数の加算
  - (浮動小数点数の乗算)
  - 誤差と丸め

誤差と丸め

- 倍精度で1と2の間で表現できる数の個数は？
  - 2^{52}-1個

  →それ以上の個数は表現できないので、
  一番近い表現で表すしかない
- 「丸める」
  - 数値を四捨五入or切り上げor切り捨てして、指定した桁に収めること
    - 四捨五入　1.4→1.0　1.5→2.0
    - 切り上げ　1.4→2.0　1.5→2.0
    - 切り捨て　1.4→1.0　1.5→1.0
  - 本来表現するべき値との誤差(丸め誤差)が発生

誤差と計算

① 計算途中保管2桁多め

- 仮数の有効桁を3桁とするとき、
  $2.26_{10}\times10^{0}+2.34_{10}\times10^{2}+1.26_{10}\times10^{0}$ はどうなるか？

途中も有効桁3桁	途中は有効桁5桁
$0.02\cancel{00}_{10}\times10^{2}$	$0.0226_{10}\times10^{2}$
$2.34\cancel{00}_{10}\times10^{2}$	$2.3400_{10}\times10^{2}$
$+\ 0.01\cancel{00}_{10}\times10^{2}$	$+\ 0.0126_{10}\times10^{2}$
$2.37\cancel{00}_{10}\times10^{2}$	$2.3752_{10}\times10^{2}$
⇩	⇩
$2.37_{10}\times10^{2}$	$2.38_{10}\times10^{2}$

計算中間式程の保管2桁 → 誤差精度

IEEE754では、計算途中では2桁余分に保持するように規定

情報落ち ③情報欠損誤差

$(-1.5_{10}\times10^{38}) + \underline{((1.5_{10}\times10^{38}) + (1))}$

$\ldots+\underline{((1.5_{10}\times10^{38}) + \mathbf{0.00\ldots}\times10^{38})}$

$=0$

絶対値の大きい数と小さい数を
足したとき、絶対値が小さい数が
無視されてしまう

$\underline{((-1.5_{10}\times10^{38}) + (1.5_{10}\times10^{38}))} + (1)$

$\underline{0}+1$

$=1$

計算の順番によって、
誤差を回避できる可能性がある

改変計算順序による誤差

桁落ち

- 浮動小数点数の演算において、有効桁数が少なくなる現象 ③有効位減少誤差
  - ごく近い数値同士の減算、計算結果が0に近くなる加減算によって発生

$(\sqrt{1000} - \sqrt{999})$ 非常に接近

$(3.1638584_{10} \times 10^{1}) - (3.1606961_{10} \times 10^{1})$ ←有効桁8桁

$= (3.1623_{10} \times 10^{-2})$ ←有効桁5桁

如何に避ける？

$= \left(\dfrac{1}{\sqrt{1000} + \sqrt{999}} \right)$ 有理化？

確認問題

- 科学記数法では、(1) × (2)(3) という形式で小数を表現する。
- IEEE754では、32ビットの(4)精度と64ビットの(5)精度がある。
- IEEE754では、計算途中は有効桁を(6)桁余分に保持している。
- 数値を丸めることによって発生する誤差を、(7)という。
- ごく近い値の浮動小数点数の減算等において、計算結果の有効桁数が少なくなる現象を(8)という。
- 絶対値の大きい数と小さい数を足したとき、絶対値が小さい数が0になってしまうことを、(9)という。

参考文献

- コンピュータの構成と設計 上 第5版
  David A.Patterson, John L. Hennessy 著、
  成田光彰 訳、日経BP社
- 山下茂 「計算機構成論 1 」講義資料