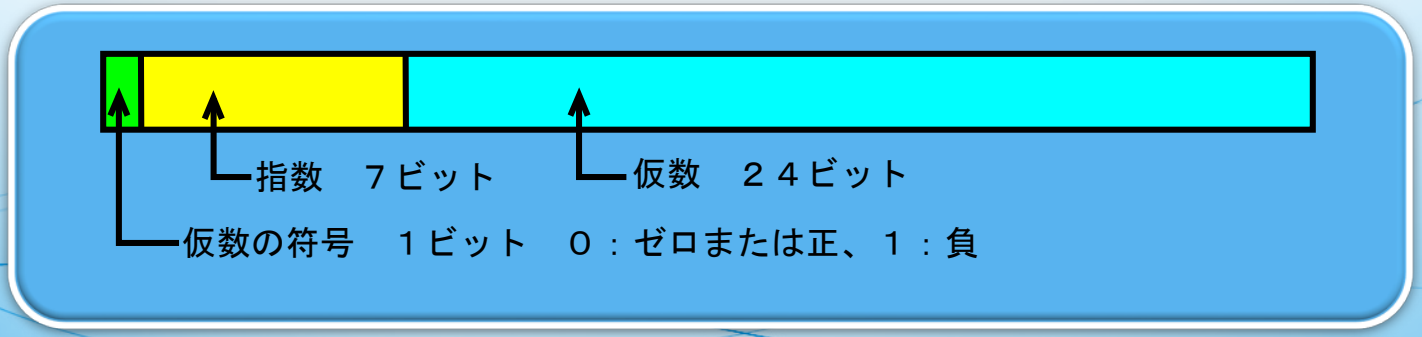
浮動小数点数

浮動小数点数

任意の大きさの実数の式

実数＝m×B^E
m：仮数
B：底（基数、通常は16または2を使用する）
E：指数



基数16の浮動小数点数の求め方

① 10進数の絶対値を2進数に変換する。
② ①で求めた2進数を正規化する。
③ 指数部の値を求める。
④ 仮数の符号を決める。
⑤ 2進数の浮動小数点数を表示する。
⑥ 2進数を16進数に変換する。

１０進数の絶対値を２進数に変換

１０進数の絶対値の整数部分、小数部分をそれぞれ２進数に変換する。

① １０進数の符号に関係なく、
１０進数の絶対値を２進数に変換する。

② 整数部分、小数部分の２進数が
それぞれ４の倍数で、かつ合計が
２４ビットになるようにビット数を調整する。

具体例

10進数の26.5を2進数に変換する。

① 整数部分の26を2進数に変換すると11010

② 小数部分の0.5を2進数に変換すると0.1

③ 2進数11010.1が求まる。

④　整数部分、小数部分を合わせて、
　　全体を4の倍数の24ビットに調整する。

000<u>11010.1</u>0000000000000000
0を付加する　　0を付加する

⑤　整数部分の上位の桁に0を付加しても値は変わらない。

⑥　小数部分の下位の桁に0を付加しても値は変わらない。

2進数の正規化

① 基数16の正規化の定義

❶ 小数点の位置を基点として、
整数部分、小数部分を
それぞれ4ビット単位にくくる

❷ 4ビット内に1を含む最上位の4ビットを、
小数点の次の4ビットになるように
4ビット単位に右又は左シフトする。

② 基数2の正規化の定義

❶ 小数第１位の桁が
最左端の有効数字になるように、

❷ ２進数のビットを
左または右にシフトすることである。

③ 正規化の結果、

❶ 基数16の場合

小数点に続く4ビットが0でない数値になる。

❷ 基数2の場合

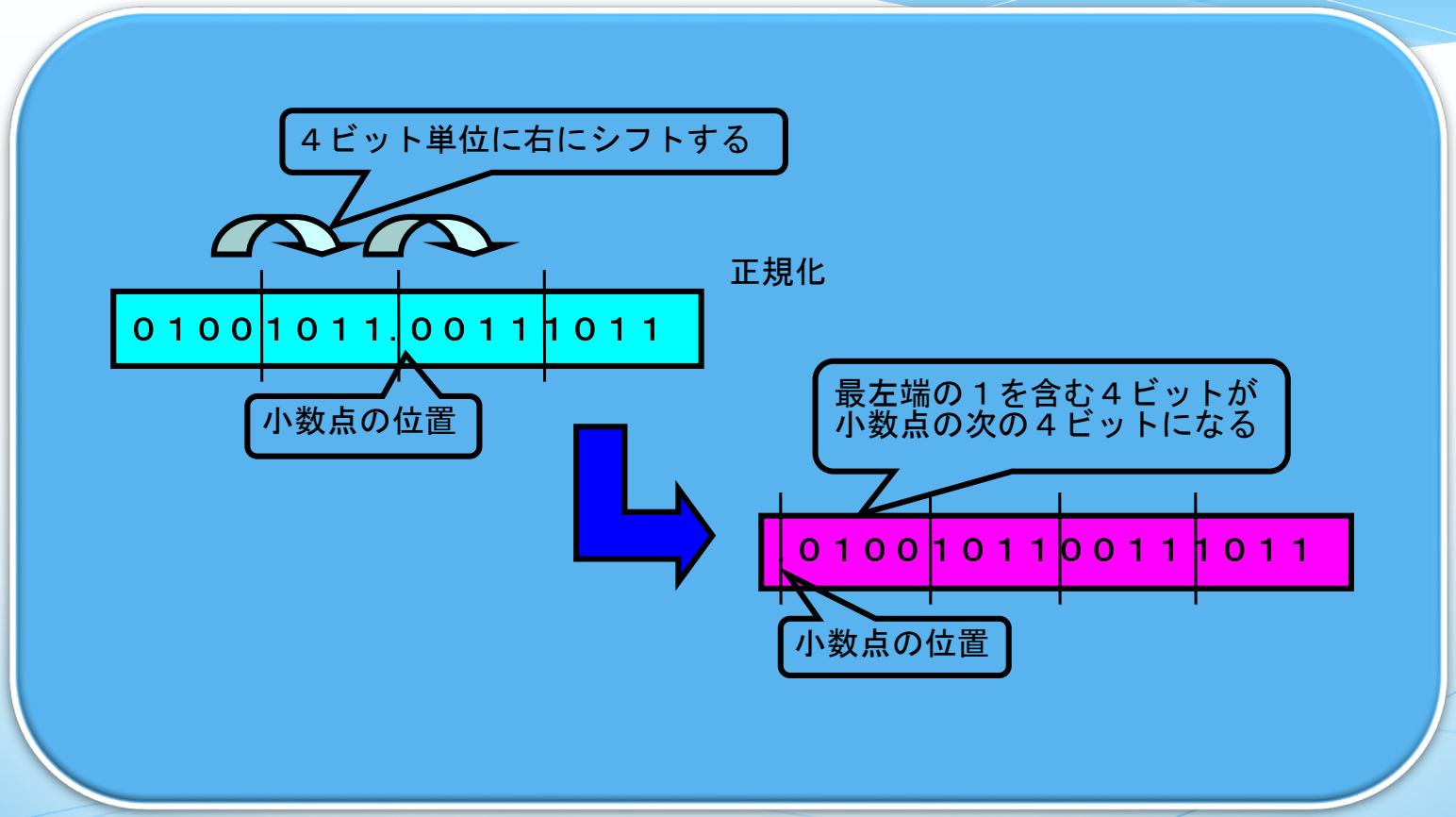
小数点の次のビットが0でない数値になる。

④ ビット数が不足する場合の処理

最下位の不足部分に0ビットを付加する。

⑤ ビットがアンダフローした場合の処理

アンダフローしたビットを捨てる。


４ビット単位に右にシフトする
０１００１０１１.００１１１０１１
小数点の位置
正規化
最左端の１を含む４ビットが
小数点の次の４ビットになる
０１００１０１１００１１１０１１
小数点の位置

具体例1

次の2進数を正規化する。

00011010.1000000000000000

① 小数点を基点にして、4ビット単位に括る。

0001 | 1010.1000 | 0000 | 0000 | 0000

② 正規化

1を含む最左端の4ビットは0001であるから、この4ビットを小数点の次の4ビットになるように、4ビット単位に右に2シフトする。すなわち、8ビット右にシフトすると次のようになる。

$.0001101010000000000000000$

③ 仮想小数点は左端となる。

④ 元の値に戻すにはこの2進数を16^2倍する必要がある。

$.0001101010000000000000000 \times 16^2$

具体例2

次の2進数の正規化

00011010.1000000000000000

① 小数点を基点にして、4ビット単位に括る。

0001 | 1010.1000 | 0000 | 0000 | 0000

② 正規化する。

1を含む最左端の4ビットは0001であるから、
この4ビットを小数点の次の4ビットになるように、
4ビット単位に右に2シフトする。

.0001101010000000000000000

③ 仮想小数点は左端となる。

④ 元の値に戻すにはこの2進数を16^2倍する必要がある。

$.0001101010000000000000000 \times 16^2$

指数部の値を決める

① 指数部の値の決め方

正規化のシフト結果を利用して
指数部の補正値の決める。

② 指数部の調整方法

❶ バイアス方式

❷ 補数表現方式

③　基数16の場合

❶　4ビット単位に右にN回シフトした場合

指数部にNを加える。

❷　4ビット単位に左にN回シフトした場合

指数部からNを減じる。

バイアス方式

① 次の計算式を使用して指数部の値を求める。

「1000000±指数部補正値」

② 指数部の値が1000000の場合

正規化された仮数の値になる。

③　指数部の補正値が＋の場合

正規化された仮数の値を16$^{(指数部の補正値)}$倍することになる。

④　指数部の補正値がーの場合

正規化された仮数の値を1／16$^{(指数部の補正値)}$倍することになる。

補数表現方式

① 次の計算式を使用して指数部の値を求める。

「OOOOOOO±指数部補正値」

② 指数部の値がOOOOOOOの場合

正規化された仮数の値になる。

③　指数部の補正値が＋の場合

正規化された仮数の値を16$^{(指数部の補正値)}$倍
することになる。

④　指数部の補正値が－の場合

正規化された仮数の値を1／16$^{(指数部の補正値)}$倍
することになる。

指数部の意味

① 基数が16の場合の指数部の増減

❶ 指数部1の増加は、仮数部の値の16倍に相当する。
❷ 指数部1の減少は、仮数部の値の16^{-1}倍に相当する。

② 基数が16の場合の仮数部のシフト結果

❶ 仮数部4ビット右にシフトすると、指数部は1だけ増加
❷ 仮数部4ビット左にシフトすると、指数部を1だけ減少

符号の決め方

仮数の符号は次の要領で設定する。

① 10進数の符号が正の場合は0

② 10進数の符号が負の場合は1

浮動小数点数の表示

浮動小数点数を次の手順で表示する。

① 符号部1ビットを2進数で表示する。

② 指数部7ビットを2進数で表示する。

③ 正規化で求めた仮数部24ビットを2進数で表示す。

但し、仮数部が16ビットの場合は16ビットに調整する。

具体例

① 問題

10進数の－125.5を、
基数16、補数表現方式の浮動小数点数に変換し、
結果を16進数で表す。

但し、符号部1ビット、指数部7ビット、
仮数部16ビットする。

②解答解説

❶ 10進数の絶対値125.5を2進数に変換する

01111101.1000

❷ 正規化する

4ビット単位に2回右にシフトして、次のようになる。

01111101. 1000 → .0111110110000000

❸ 指数部は次のようになる。

0000000＋0000010＝0000010

❹ 符号部は10進数が負であるから1となる。

❺ 2進数の表示

10000010011111011000000

❻ 16進数に変換

827D80

基数2の正規化

① 基数2の正規化は
仮数部の最上位桁が0にならないように
指数部と仮数部を調整する操作である。

② 最左端の1のビットが
小数点の次のビットの小数1位になるように、
2進数を左にシフトまたは右にシフトする操作である。

③ 浮動小数点数を求める場合、
基数16と基数2の場合の違いは正規化の操作である。

基数2の指数部の値の変化

① 基数が2の場合の指数部の増減

❶ 指数部1の増加は、仮数部の値の2倍に相当する。
❷ 指数部1の減少は、仮数部の値の2^{-1}倍に相当する。

② 基数が2の場合の仮数部のシフト結果

❶ 仮数部1ビット右にシフトすると、指数部を1だけ増加
❷ 仮数部1ビット左にシフトすると指数部を1だけ減少