２進数の計算と数値表現

基本情報技術者　第二章

ルーツ千葉　利用者
長岡　昇吾

2進数の足し算

10進数	$\begin{array}{r} 163 \\ +\ \ 46 \\ \hline 209 \end{array}$ ＋1	$\begin{array}{r} 163 \\ -\ \ 46 \\ \hline 117 \end{array}$ －1
2 進数	$\begin{array}{r} 111 \\ +\ \ 10 \\ \hline 1001 \end{array}$ ＋1　＋1	$\begin{array}{r} 101 \\ -\ \ 11 \\ \hline 10 \end{array}$ －1

コンピュータには「引き算」という概念がない
（引き算用の回路を組んだ場合では、今後紹介する方法と比べて効率が悪くなるため）

2進数で負の数を表す

0 0 0 0 0 0 1 1
1 0 0 0 0 0 1 1

とします

0なら正
1なら負

ただ、二つの数を足すと…

1 0 0 0 0 1 1 0

何故か0になりません

2進数で負の数を表す

足して０になる数字を作って負の数とすればよい

０ ０ ０ ０ ０ ０ １ １ ＋χ
＝ ０ ０ ０ ０ ０ ０ ０ ０

オーバーフローさせれば先頭の１が消えるので足して０にできる。

０ ０ ０ ０ ０ ０ １ １
＋ １ １ １ １ １ １ ０ １
＝ １ ０ ０ ０ ０ ０ ０ ０ ０

補数(１と０が反対になっている数字)＋１じゃない？

χ＝ １ １ １ １ １ １ ０ １

2進数法におけるシフト

１０進法の場合

6970000

10進法で大きな値を簡単にするには

6.97×10^{5}

２進法の場合

１１０１１１００００

この２進法であらわされた数字を簡単に直すと

110111×2^{4}

2進数法におけるシフト

１０進法の場合

0.0026

2.6×10^{-3}

10進法で大きな値を簡単にするには

２進法の場合

１１０１１１

１１０１１１０００ $\times 2^{-3}$

＝

１１０１１１０００ $\times 1/2^3$

この２進法であらわされた数字を簡単に直すと

理論シフトと算術シフト

理論シフト (符号は考慮しない)

1	0	1	0	1	1	0	0

$\times 2^{2}$

1	0	1	0	1	1	0	0	0	0

1	0	1	0	1	1	0	0

$\times 1/2^{2}$

0	0	1	0	1	0	1	1	0	0

算術シフト (符号を考慮する)

1	0	1	0	1	1	0	0

$\times 2^{2}$

1	0	1	0	1	1	0	0	0	0

1	0	1	0	1	1	0	0

$\times 1/2^{2}$

1	1	1	0	1	0	1	1	0	0

2進数の掛け算と割り算

2進数

１ １×１ １ １＝χ

２＋１

2^2+2+1

$$
\begin{array}{rrl}
 & 111 & \times 2 \\
+ & 111 & \times 1 \\
\hline
 & 10101 &
\end{array}
$$

2進数

１ ０ １ １ １÷１ １＝χ

$$
\begin{array}{rrl}
 & 10111 & \\
- & 11 & \times 2^2 \\
\hline
 & 1011 &
\end{array}
$$

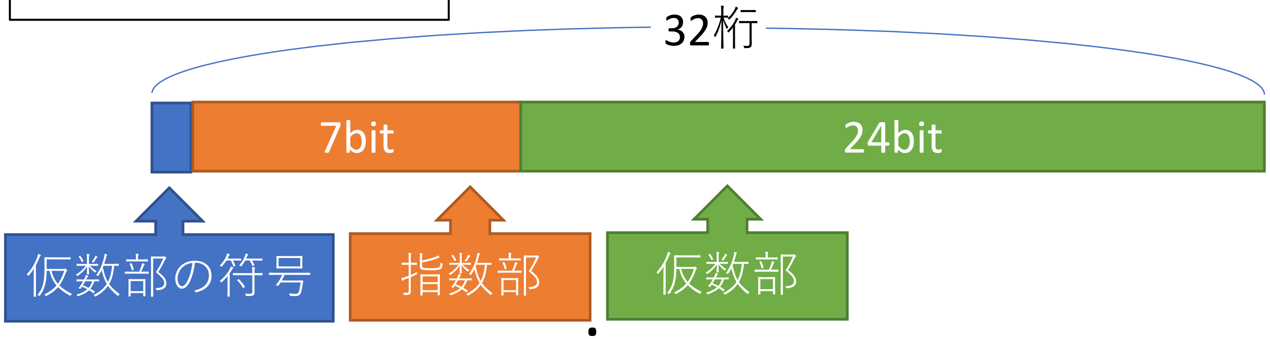
$$
\begin{array}{rrl}
 & 1011 & \\
- & 11 & \times 2 \\
\hline
 & 101 &
\end{array}
$$

$$
\begin{array}{rrl}
 & 101 & \\
- & 11 & \times 1 \\
\hline
 & 10 &
\end{array}
$$

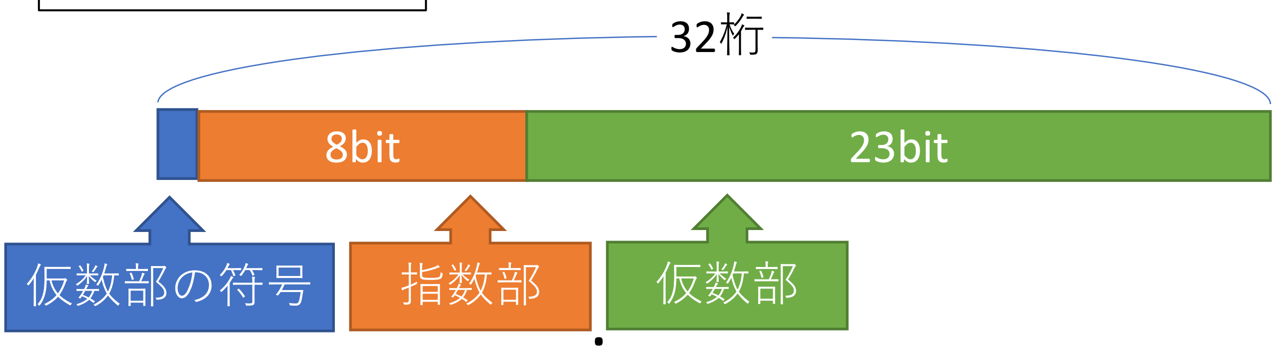
x＝１ １ １・・・１ ０

2進数の固定小数点と浮動小数点

32ビット形式



IEEE754形式



浮動小数点数

0.11×2^{3}

32ビット形式で表示すると指数部３を7bit
に表して、小数点の後ろから表示する

バイアス値

実際の指数		バイアス		指数		2進数
-127		１２７		０		00000000
０	＋	１２７	＝	127	→	01111111
128		１２７		255		11111111

マイナスの値を表す為にしている

コンピュータ内では

誤差

無限小数などを扱う際に「極力近い値」で済ませてあげる必要が出てきます。

けたあふれ誤差

0	0	0	0	0	0	0	0	1

情報落ち

0 1 1 0 ・・・ 0 0 1 0 $\xrightarrow{\text{正規化}}$ **0.011**

打切り誤差

0 1 1 0 ・・・ 0 ~~0 1 0~~

けた落ち

信用できる桁数で区切るものです。

8.93-8.885=0.04~~5~~　(偶数丸めを採用)

丸め誤差

[JIS Z 8401:1999 数値の丸め方 (kikakurui.com)]

あふれた桁数に対して四捨五入や切り上げをした際に出る誤差

問題の解

問1
普通に計算しましょう。
１１０１００００
⇩右に２ビットシフト
１１１１０１００００
⇩引き算は補数を取って１を足せばよいので
０００１０１００ + （- １１１１０１００）
⇩
０００１０１００＋（００００１０１１+０００００００１）
⇩
０００１０１００+００００１１００
⇩
００１０００００

A. ００１０００００

問題の解

10進数	A	B	C	D
	１０	１１	１２	１３

2進数	A	B	C	D
	1010	1011	1100	1101

２ビットシフト

2進数	2	A	F	3
	0010	1010	1111	0011

A.2AF3

問題の解

問３

左２ビットシフトは $x \times 2^2$ なので４倍
もう一度 x を足すので４ x ＋ x なので５倍

A. 5

問４

b1～bn＝Aと置く
⇩
３Aを２進数に直すと11×A
⇩
b1～bn０ ＋ b1～bn

A. b1～bn０ ＋ b1～bn

問題の解

問１

－5.625を２進数に直します

⇩

-(2^2 ＋ １ ＋1/2＋1/2^3)

⇩枠の中に納める

-(０ １ ０ １ . １ ０ １ ０)

⇩マイナスにする

１ ０ １ ０． ０ １ ０ １ + ０ ０ ０ ０． ０ ０ ０ １

⇩

１ ０ １ ０． ０ １ １ ０

A. １ ０ １ ０． ０ １ ０ ０

問題の解

問２

10進数の0.25を要素ごとに分けます
S：符号(０正１負)
e：指数部分(補数表現で4bit)
ｆ：仮数部分(絶対値で11bit)
Sはまず正のため０
0.25は$２^{－２}$と表せるので
$0.1\times ２^{－１}$
eは０００１の補数＋１して１１１１
ｆは１０００００００００
合わせて
０１１１１１００００００００００

A.ウ

過去問

問１

表示解像度が1000×800ドットで、色数が65536色(2の16条)の画像を表示するのに最低限必要なビデオメモリ容量は何Mバイトか。

１M＝１０００ｋバイト　　１ｋバイト＝１０００バイトとする

ア　1.6　　イ　3.2　　ウ　6.4　　エ　12.8

まず１０００×８００の１マスに２^{16}の要素が必要であると考えてください,
この画像を表すのに１０００×８００×１６桁必要であることがわかると思います。
１M＝１０００×１０００バイトで,
１バイト＝８bitなので
0.1×16＝1.6Mバイトとなるため,
答えはアとなる

過去問

問２

　次に示す手順は，列中の少なくとも一つは１ビットであるビット列が与えられたとき，最も右にある１を残し他のビットを全て０にするアルゴリズムである。例えば00101000が与えられたとき，00001000が求まる。aに入る理論演算はどれか。

手順1　与えられたビット列Aを符号なしの２進数と見なし，Aから１を引き，結果をBとする
手順２　AとBの排他的論理和(XOR)を求め，結果をCとする。
手順３　AとCの（　a　）を求め結果をAとする

ア　排他的論理和(XOR)　イ　否定的論理積(NAND)
ウ　論理積(AND)　エ　論理和(OR)

まずはBに入ってくる数字を考えましょう。
Aの値を１１１０００００として最初の2桁の11をｘ ３～８桁をｙとします。
Aから１を引いたBを見てみるとｘ’＝11、ｙ’＝011111となることに気づきませんか？
Ｃは排他的論理和を使っているため１と０または０と１の時に１を返します。
Ｃを同じように見るとｘ”＝00、ｙ”＝111111になります。
もしも、アを選ぶとｘ”’＝11、ｙ”’＝011111、イを選ぶとｘ”’＝11、ｙ”’＝011111
ウを選ぶとｘ”’＝00、ｙ”’＝100000、エを選ぶとｘ”’＝11、ｙ”’＝111111、
よって答えはウ