位取り基数法(2進数，10進数，16進数)

山本昌志*

2005年10月21日

1　本日の学習内容

教科書のp.5〜12の内容を本日，学習する．ここの内容は，2年生の時に学習したはずである．したがって，本日の講義は，ほとんど2年生の復習となる．なかには，2年生の時の内容を完全にマスターしている者も居るだろう．そういう者向けに，少しだけ高度な内容も載せているので，それを理解して欲しい．

アセンブラ言語を理解するためには，メモリーを強く意識しなくてはならない．これがよく分かるためには，プログラムを構成するデータや命令の表現方法を理解しなくてはならない．諸君の大部分の者は，コンピューター内部では2進数が使われていることを知っていると思う．コンピューターの内部ではデータや命令は全て2進数で表現されているのである[1]．

2進数で命令を表現することは後の講義で示すことになる．ここでは，整数を2進数で表すことを学習する．16進数は2進数の親戚で，表示に便利なのでコンピューターの世界でよく使われる．そのため，16進数も学習する．

ところで，コンピューターで2進数が使われる理由は，なんであろうか?．それについては，付録の5.1に示してある．

2　位取り記数法

2.1　数の表現

通常使われている10進数では，0〜9までの10個の数字を使って，数を表現する．9の次は10で桁が上がる．10進数以外にいろいろな数の数え方がある．2進数，10進数，16進数の数の表現を図1に示す．合わせて，桁上がりという考えの無い漢字や楔形文字，ローマ数字も併記する．桁上がりという考えが無い表記の場合，桁が上がるたびに，新しい記号が必要であることがわかる．大きな数字を表す場合，非常に不便である．また，筆算を用いた計算もできない．

*独立行政法人　秋田工業高等専門学校　電気工学科

[1]この辺の話は，アセンブラ言語を理解すると分かってくるはずである

原始人	2進法	10進法	16進法	漢字	楔形文字	ローマ数字
	0	0	0			
●	1	1	1	一	▼	I
●●	10	2	2	二	▼▼	II
●●●	11	3	3	三	▼▼▼	III
●●●●	100	4	4	四	▼▼▼▼	IV
●●●●●	101	5	5	五	▼▼▼▼▼	V
●●●●●●	110	6	6	六	▼▼▼▼▼▼	VI
●●●●●●●	111	7	7	七	▼▼▼▼▼▼▼	VII
●●●●●●●●	1000	8	8	八	▼▼▼▼▼▼▼▼	VIII
●●●●●●●●●	1001	9	9	九	▼▼▼▼▼▼▼▼▼	IX
●●●●●●●●●●	1010	10	A	十	◀	X
●●●●●●●●●●●	1011	11	B	十一	◀▼	XI
●●●●●●●●●●●●	1100	12	C	十二	◀▼▼	XII
●●●●●●●●●●●●●	1101	13	D	十三	◀▼▼▼	XIII
●●●●●●●●●●●●●●	1110	14	E	十四	◀▼▼▼▼	XIV
●●●●●●●●●●●●●●●	1111	15	F	十五	◀▼▼▼▼▼	XV
●●●●●●●●●●●●●●●●	10000	16	10	十六	◀▼▼▼▼▼▼	XVI
●●●●●●●●●●●●●●●●●	10001	17	11	十七	◀▼▼▼▼▼▼▼	XVII
●●●●●●●●●●●●●●●●●●	10010	18	12	十八	◀▼▼▼▼▼▼▼▼	XVIII
●●●●●●●●●●●●●●●●●●●	10011	19	13	十九	◀▼▼▼▼▼▼▼▼▼	XIX

図 1: 数の数え方

2.2 現代の数の表現 (位取り記数法)

現代の便利な数の表現は桁上がりの考えがあるからこそである．この桁上がりの考えは，ゼロが発見されたので可能となった．このように桁上がりの考えで，数を示すのが位取り記数法 (place value sysytem) である．10 進数は 9 の次で桁上がりが生じ 10 となる．0〜9 の数字を使って，数を表すのである．この 10 を基数と，0〜9 間での数を底と言う．10 進数の他，いろいろの基数の数が考えられるが，コンピューター科学で使われるのは，主に 2 進数と 16 進数である．それぞれの基数と底を表 1 に示す．

表 1: 基数と底

数の表現	基数	底
2 進法	2	0, 1
10 進法	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
16 進法	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

図 1 や表 1 から，自然数の数え方は理解できたと思う．そうすると相互の変換ができれば，ある程度それを応用することができるようになる．それぞれの変換を考える前に，数の表記方法について，勉強することにする．位取り記数法での数の表し方が理解できれば，それぞれの変換が分かるはずである．

例えば，今年は 2005 年である．10 進法の 2005 の表記はどのような意味があるか?．これは，次のように解釈する．大げさではあるが，こう解釈すると，他の基数の数字の意味はっきりする．

$$(2005)_{10} = (2 \times 10^3 + 0 \times 10^2 + 0 \times 10^2 + 5 \times 10^0) \tag{1}$$

括弧の下の 10 は 10 進法の意味で，非常に簡単な話である．これさえ，分かれば基数の変換なんか，怖くない．この式の右辺の $\times$ と 10^p をとって，それを並べたのが位取り記数法である．

コーヒーブレイク

ゼロがない時代は，大変だった．ゼロが無いと，式 (16) の左辺のような表記は出来ない．その 0(ゼロ) が発見されたのは，6 世紀頃のインドと言われている．西暦 0 年がないのは，このためである．キリストが生まれた頃は，ゼロがなかったのである．

3 基数の変換

3.1 2 進数と 10 進数の変換

3.1.1 2 進数 → 10 進数

2 進数から 10 進数への変換は簡単である．式 (16) を理解していれば，分かる．2 進数であろうが 10 進数であろうが，表記法は同じで，約束に従って変形すれば良い．次のようにする．

$$\begin{aligned}
(10011)_2 &= (1 \times 10^{100} + 0 \times 10^{11} + 0 \times 10^{10} + 1 \times 10^1 + 1 \times 10^0)_2 \\
&= (1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0)_{10} \\
&= (16 + 0 + 0 + 2 + 1)_{10} \\
&= (19)_{10} \qquad (2)
\end{aligned}$$

この手中を順を追って説明すると，以下のようになる．

1. まずは，1 行目右辺のように位取り記数法で表現する．
2. そうして，表 1 に従い，式を変換したい基数に直す．
3. 後は地道に計算するだけ．

通常は，1 行目の右辺は省き，2 行目から計算する．

- 2 進数の各桁の 10 進数での値 (重み) を覚えておくと便利である．
  1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096

3.1.2　10 進数 → 2 進数

今度は逆で 10 進数から 2 進数への変換である．原理的に，先ほどと同じように変換ができるが，計算してみるとそれは難しい[2]．これは，我々は 10 進数の演算になれていることが原因となっている．自然には，10 進数であろうと 2 進数であろうと優位性は無いからである．

10 進数から 2 進数への計算手法は簡単であるが，その内容を理解することが大事である．計算手法を忘れても，内容が理解できていれば，その方法はいつでも自分で作ることができる．また，応用範囲も広がる．では，簡単な例で説明する．10 進数の $(19)_{10}$ を 2 進数に変換する方法を示す．具体的には，19 を

$$(19)_{10} = (\cdots + a_4 \times 2^4 + a_3 \times 2^3 + a_2 \times 2^2 + a_1 \times 2^1 + a_0 \times 2^0)_{10} \tag{3}$$

と表現したい．これは式 (2) の 2 行目の式で，ここで求められた係数を $(\cdots a_4 a_3 a_2 a_1 a_0)_2$ と並べれば位取り記数法になる．それぞれ，a_n を求めなくてはならない．そこで，次のように 19 を 2 で割った商と余りを考える．これは，

$$(9 \times 2 + 1)_{10} = (\cdots + a_4 \times 2^3 + a_3 \times 2^2 + a_2 \times 2^1 + a_1 \times 2^0)_{10} \times 2 + a_0 \tag{4}$$

と書ける．これをよくにらむと，$a_0 = 1$ ということが分かる．すなわち，a_0 は 19 を 2 で割ったあまりである．残りの部分は，

$$(9)_{10} = (\cdots + a_4 \times 2^3 + a_3 \times 2^2 + a_2 \times 2^1 + a_1 \times 2^0)_{10} \tag{5}$$

となることも分かるでろう．商について同じことをすると，

$$(4 \times 2 + 1)_{10} = (\cdots + a_5 \times 2^3 + a_4 \times 2^2 + a_3 \times 2^1 + a_2 \times 2^0)_{10} \times 2 + a_1 \tag{6}$$

となる．したがって，$a_1 = 1$ である．しつこいが，さらに商について同様に進めると，

$$(2 \times 2 + 0)_{10} = (\cdots + a_6 \times 2^3 + a_5 \times 2^2 + a_4 \times 2^1 + a_3 \times 2^0)_{10} \times 2 + a_2 \qquad \Rightarrow \qquad a_2 = 0 \tag{7}$$
$$(1 \times 2 + 0)_{10} = (\cdots + a_7 \times 2^3 + a_6 \times 2^2 + a_5 \times 2^1 + a_4 \times 2^0)_{10} \times 2 + a_3 \qquad \Rightarrow \qquad a_3 = 0 \tag{8}$$
$$(0 \times 2 + 1)_{10} = (\cdots + a_8 \times 2^3 + a_7 \times 2^2 + a_6 \times 2^1 + a_5 \times 2^0)_{10} \times 2 + a_4 \qquad \Rightarrow \qquad a_4 = 1 \tag{9}$$

となる．最後の式から，$a_n = 0 \quad (5 \leq n)$ が分かる．以上をまとめると

$$\begin{aligned}(19)_{10} &= (a_4 \times 2^4 + a_3 \times 2^3 + a_2 \times 2^2 + a_1 \times 2^1 + a_0 \times 2^0)_{10} \\ &= (1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0)_{10} \\ &= (10011)_2\end{aligned} \tag{10}$$

となる．要するに，2 で割ったあまりを書いていけば良いのである．計算方法は分かった．だがこの方法は実際的ではない．よく使われるのは，図 2 のように 2 で割った余りを並べる．これは，$(19)_{10} = (10011)_2$，　$(2003)_{10} = (11111010011)_2$ を示している．

[2]本当に難しいか，試して見よ

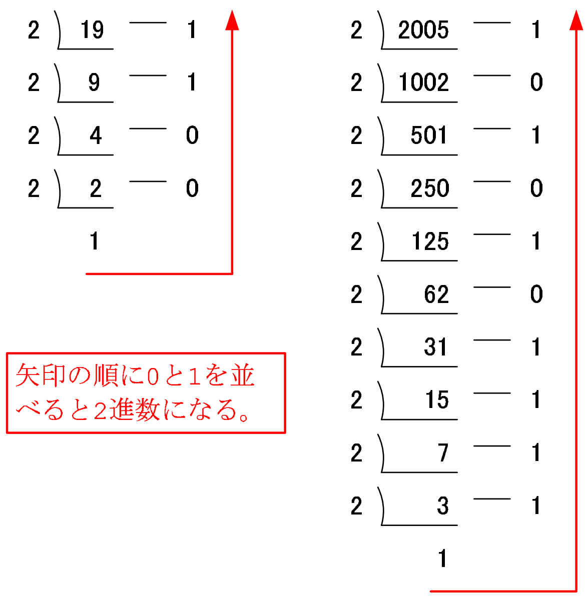

図 2: 10 進数から 2 進数への変換方法.

内容を十分理解し，変換の練習をしなくてはならない.

3.2　16 進数

2 進数の変換が理解できたら，16 進数の変換はまったくもって簡単である.

3.2.1　16 進数 → 10 進数

2 進数と同じで次のようにする.

$$\begin{aligned}(376)_{16} &= (3\times 10^2+7\times 10^1+6\times 10^0)_{16}\\ &= (3\times 16^2+7\times 16^1+6\times 16^0)_{10}\\ &= (3\times 256+7\times 16+6\times 1)_{10}\\ &= (886)_{10}\end{aligned} \tag{11}$$

3.2.2　10 進数 → 16 進数

これも 2 進数と同様に考える．16 で割ったあまりを並べれば良い．図 3 のようにして，$(25391)_{10} = (632F)_{16}$ を計算する.

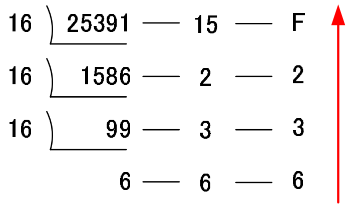

図 3: 10 進数から 16 進数への変換方法

手計算ではこの方法を用いるが，実際のエンジニアーは電卓の変換機能を使う．

コーヒーブレイク

昔から言われるジョークをひとつ．プログラマーは，クリスマス (12 月 25 日) とハロウィーン (10 月 31 日) が区別できない．なぜか?．ヒント

- 10 進数 (decimal number) のことを DEC と書く．DEC 23 と書けば，10 進数の 23 をあらわしている．
- 8 進数 (octal number) はのことを OCT とかく．OCT 23 と書けば，8 進数の 23 を表している．

3.3 2 進数と 16 進数の変換

2 進数と 16 進数の相互の変換は簡単である．$2^4 = 16$ なので，2 進数の 4 桁は 16 進数の一桁に対応している．図 4 のように，1 桁の 16 進数を 4 桁の 2 進数に変換すれば良い．反対に 4 桁の 2 進数は，1 桁の 16 進数に変換できる．

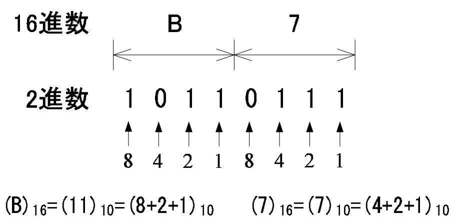

図 4: 2 進数と 16 進数の相互変換方法

4 課題 (レポート)

4.1 記数法の変換

以下の練習問題を解き，レポートとして提出すること．

[問 1] 2 進数と 16 進数へ変換せよ.

$(1)_{10}$	$(2)_{10}$	$(4)_{10}$
$(8)_{10}$	$(16)_{10}$	$(32)_{10}$
$(65536)_{10}$	$(2004)_{10}$	$(999)_{10}$

[問 2] 以下の 2 進数を 10 進数と 16 進数へ変換せよ.

$(1)_2$	$(10)_2$	$(100)_2$
$(1000)_2$	$(11111)_2$	$(10101111)_2$
$(10010101)_2$	$(10101010)_2$	$(11111111)_2$

[問 3] 以下の 16 進数を 2 進数と 10 進数へ変換せよ.

$(8)_{16}$	$(F)_{16}$	$(1F)_{16}$
$(AF)_{16}$	$(F98)_{16}$	$(89AB)_{16}$
$(CDEF)_{16}$	$(FFFF)_{16}$	$(A000)_{16}$

4.2 レポート提出要領

提出方法は，次の通りとする.

期限	11 月 2 日 (金)AM8:55 まで
用紙	A4
提出場所	山本研究室の入口のポスト
表紙	表紙を 1 枚つけて，以下の項目を分かりやすく記述すること. 授業科目名「電子計算機」 課題名「課題 2　基数の変換」 3E　学籍番号　氏名 提出日
内容	問題の解答．計算課程をきちんと書くこと.

5　付録

5.1　コンピューターで 2 進数が使われる理由

5.1.1　2 進数のメリット

人間の指は 10 本あることは，よく知られている．そのため，人類は 10 進法を使っていると言われている．小学校の低学年では指を使って計算する子供がいることからも分かる．コンピューターの内部のハードウェアーでは，電圧が 0V か 5V(もっと低い場合もある) でデータやプログラムを表現している．指が 2 本しかないのと同じ．だから，コンピューターは 2 進法を使う．2 進法を使うメリットに，何があるか? という疑問が湧くであろう．その答えとして，以下のようなことが考えられる．

- ノイズに強い

  0〜5V で動作する素子からできたコンピューターを考える．2 ビットと 10 ビットの場合，割り当てられる電圧のレベルは，図 5 の通りである．図からも分かるように，許されるノイズは，2 ビットの方が格段に大きくなる．1 ビットのエラーも許されないデジタルコンピューターにおいては，この差は非常に大きい．

- ハードウェアーを実現するのが容易

  コンピューター内部には，単純な動作をする同じような部品が数多くある．2 進数であれば，入力は 0 と 1，出力も 0 と 1 なので，構成する 1 個の部品が非常に単純になる．要するに 2 進数を採用すると，部品が簡単になるのである．

- 演算が簡単

  例えば，掛け算九九を考えると分かる．10 進数だと，0〜9 までの掛け算，合計 100 通りある．2 進数だと，4 通りで済む．

- ブール代数が使えて，論理演算が容易

  ブール代数については，他の授業で勉強することになっている．それまで待てない人は，私の講義ノート[3]でも見ろ．

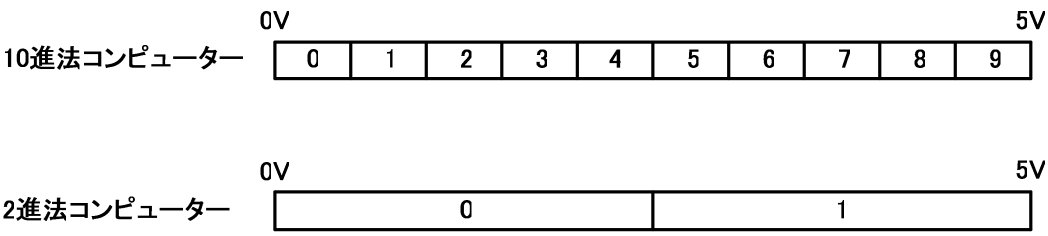

図 5: 2 進法と 10 進法のコンピューターのノイズレベル

[3] http://www.akita-nct.jp/ yamamoto/lecture/2003/2E/boolean_algebra/index.html

5.1.2　2 進数のデメリット

それでは，2 進数のデメリットとはどのようなことが考えられるであろうか?．すぐに分かることは，桁数が多くなることである．例えば，十進法の $(99)_{10}$ は，二進法では $(1100011)_2$ となる．十進法であれば 2 桁で済むのに，2 進法であれば 7 桁も必要になる．

このことは，コンピューターが発明された当初問題とされたが，すぐにこれは間違いだと気づく．ある正の整数 k をそれぞれの位取記数法で表した場合，その底の数 α と桁数 β を表 2 にしめす．n 進数の場合，整数 k を表すの必要な桁数 $\log_n k$ は次のようにして理解できる．n 進数が β 桁あると，それが表すことができる組み合わせの数は，n^β となる．これが，s 整数 k まで表すことができるから，

$$n^\beta = k \tag{12}$$

となる．したがって，必要な桁数は，

$$\beta = \log_n k \tag{13}$$

となる．

表 2: 整数 k を表すときの底と桁数

底の数 α	桁数 β
1	k
n	$\log_n k$
k	1

それでは，一番効率のよい底の数はいくつであろうか?．効率の定義はいろいろ出来るが，ここでは

- 底の数と桁数で評価することにする．ある整数を表す場合，これらの数の合計が小さいことが，効率がよい．

とする．n 進数で正の整数 k を表す場合，効率は，

$$\alpha + \beta = n + \log_n k \tag{14}$$

となる．これが最小となるのは，n で微分したときゼロとなる，

$$\begin{aligned}\frac{\mathrm{d}(\alpha+\beta)}{\mathrm{d}n} &= 1 - \frac{\log k}{n(\log n)^2} \\ &= 0 \end{aligned} \tag{15}$$

である．この方程式の解，すなわちもっとも効率の良い底を図 6 に示す．この図から分かるように，比較的小さな数字 ($\leq 10^5$) では 4 進数あたりが効率がよい．大きな数になると 10 進数程度が最適な底となる．コンピューターが扱う最大の数は，大体 2^{32} 程度[4]である．これだと，2 進数で 32 桁，10 進数で 10 桁である．

[4] C 言語の int 型の最大値

この場合，図から分かるようにもっとも効率の良いのが 6 進数であるが，これだと，13 桁が必要である．2 進数，6 進数，10 進数をつかっても，桁数は 10〜32 である．コンピューターにとって，32 桁を取り扱うことは簡単なことなので，2 進数で数字を表現しても全く問題ない．10 進数をつかうことに比べて 3 倍程度の桁数の増加にしかすぎないのである．2 進数を使うことによる桁数の増加は，さほど大きなデメリットではない．それよりも，2 進数を使うメリットの方が圧倒的に多い．

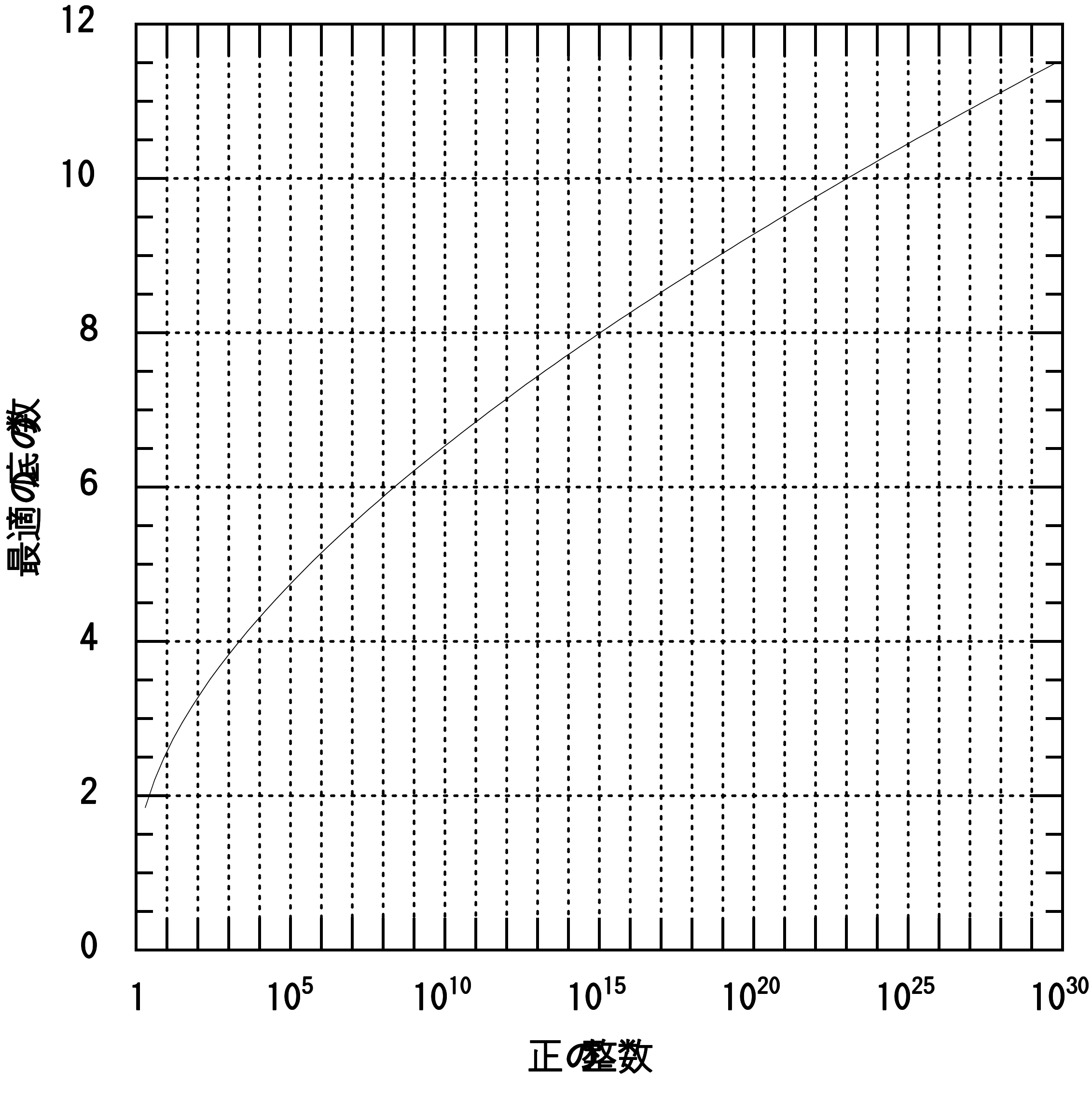

図 6: もっとも効率の良い底．横軸は整数で，縦軸はその整数を表すときのもっとも効率の良い底．

5.2　少数の表現 (おまけ)

5.2.1　位取り記数法

10 進数での小数の表現を考える。例えば，小数の表現,

$$(0.1235)_{10} = (1 \times 10^{-1} + 2 \times 10^{-2} + 3 \times 10^{-3} + 5 \times 10^{-4}) \tag{16}$$

と整数の場合と同じようになっている。小数点を境に、右側の指数部が-1, -2, -3 と 1 ずつ減少する。これは、先に示した整数の場合と全く同じで，簡単である。当然,

$$\begin{aligned}
10^{-1} &= \frac{1}{10^1} = \frac{1}{10} = 0.1 \\
10^{-2} &= \frac{1}{10^2} = \frac{1}{100} = 0.01 \\
10^{-3} &= \frac{1}{10^3} = \frac{1}{1000} = 0.001 \\
&\vdots \\
10^{-N} &= \frac{1}{10^N}
\end{aligned} \tag{17}$$

は理解していなくてはならない.

5.2.2　基数の変換 (2 進数 →10 進数)

2 進数での少数の表記も、10 進数の場合と同じである。だから、2 進数少数を 10 進数少数に変換するのは簡単である。たとえば、

$$\begin{aligned}
(0.10101)_2 &= (1 \times 10^{-1} + 0 \times 10^{-10} + 1 \times 10^{-11} + 0 \times 10^{-100} + 1 \times 10^{-101})_2 \\
&= (1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 0 \times 2^{-4} + 1 \times 2^{-5})_{10} \\
&= (1 \times 0.5 + 0 \times 0.25 + 1 \times 0.125 + 0 \times 0.0625 + 1 \times 0.03125)_{10} \\
&= (0.5 + 0.125 + 0.03125)_{10} \\
&= (0.65625)_{10}
\end{aligned} \tag{18}$$

となる．当然

$$\begin{aligned}
2^{-1} &= \frac{1}{2^1} = \frac{1}{2} = 0.5 \\
2^{-2} &= \frac{1}{2^2} = \frac{1}{4} = 0.25 \\
2^{-3} &= \frac{1}{2^3} = \frac{1}{8} = 0.125 \\
&\vdots \\
2^{-N} &= \frac{1}{2^N}
\end{aligned} \tag{19}$$

は理解していなくてはならない.

5.2.3 基数の変換 (10 進数 →2 進数)

つぎは，先ほどと逆を考える．たとえば，先ほどの例の $(0.65625)_{10}$ を 2 進数で表現する．そのためには，

$$(0.65625)_{10} = (a_1 \times 2^{-1} + a_2 \times 2^{-2} + a_3 \times 2^{-3} + \cdots)_{10} \tag{20}$$

と書き直せばよい．ただし，a_n は 0 または 1 である．そして，この a_n を並べると，

$$(0.65625)_{10} = (a_1 a_2 a_3 \cdots a_4)_2 \tag{21}$$

と 2 進数で表現できる．ここで，問題は a_n を求めることである．そこで，式 20 の両辺を 2 倍する．すると，

$$(1.3125)_{10} = (a_1 \times 2^0 + a_2 \times 2^{-1} + a_3 \times 2^{-2} + \cdots)_{10} \tag{22}$$

となる．この式の両辺の整数部と小数部は等しいので，

$$(1)_{10} = (a_1)_{10} \qquad (0.3125)_{10} = (a_1 \times 2^0 + a_2 \times 2^{-1} + a_3 \times 2^{-2} + \cdots)_{10} \tag{23}$$

となる．これで，$a_1 = 1$ が求まった．同じように，残りの小数部分を 2 倍すると，

$$(0.625)_{10} = (a_2 \times 2^0 + a_3 \times 2^{-1} + a_4 \times 2^{-2} + \cdots)_{10} \tag{24}$$

となる．これも，両辺の整数部と小数部が等しいので，

$$(0)_{10} = (a_2)_{10} \qquad (0.625)_1 0 = (a_3 \times 2^{-1} + a_4 \times 2^{-2} + a_5 \times 2^{-3} + \cdots)_{10} \tag{25}$$

が得られる．これで，$a_2 = 0$ が求まった．同様に以下の通り，残りの小数部分の計算を進めると，全ての a_n が求まる．

$$\begin{cases} (1.25)_{10} = (a_3 \times 2^{-1} + a_4 \times 2^{-2} + a_5 \times 2^{-2} + \cdots)_{10} \\ (1)_{10} = (a_3)_{10} \qquad (0.25)_{10} = (a_4 \times 2^{-1} + a_5 \times 2^{-2} + a_6 \times 2^{-2} + \cdots)_{10} \end{cases} \tag{26}$$

$$\begin{cases} (0.5)_{10} = (a_4 \times 2^{-1} + a_5 \times 2^{-2} + a_6 \times 2^{-2} + \cdots)_{10} \\ (0)_{10} = (a_4)_{10} \qquad (0.5)_{10} = (a_5 \times 2^{-1} + a_6 \times 2^{-2} + a_7 \times 2^{-2} + \cdots)_{10} \end{cases} \tag{27}$$

$$\begin{cases} (1.0)_{10} = (a_5 \times 2^{-1} + a_6 \times 2^{-2} + a_7 \times 2^{-2} + \cdots)_{10} \\ (1)_{10} = (a_5)_{10} \qquad (0.0)_{10} = (a_6 \times 2^{-1} + a_7 \times 2^{-2} + a_8 \times 2^{-2} + \cdots)_{10} \end{cases} \tag{28}$$

最後に、小数部がゼロとなったので計算は、完了となる。以上をまとめると

$$\begin{aligned} (0.65625)_{10} &= (a_1 \times 2^{-1} + a_2 \times 2^{-2} + a_3 \times 2^{-3} + \cdots)_1 0 \\ &= (1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 0 \times 2^{-4} + 1 \times 2^{-5} +)_1 0 \\ &= (0.10101)_2 \end{aligned} \tag{29}$$

となる．要するに、小数部を 2 倍して、その整数部を書いていけばよい．

よく使われるのは、図 7 のようにして計算を進める．2 倍して，整数部を書き出して，小数部を再度 2 倍する．これを繰り返すと，10 進数小数を 2 進数小数に変換することができる．10 進数の 0.1 は循環小数ではないが、2 進数にすると、

$$(0.1)_{10} = (0.00011001100110011 \cdots)_2 \tag{30}$$

と循環小数になる．通常は，途中まで (必要な精度まで) で，計算を打ち切る．

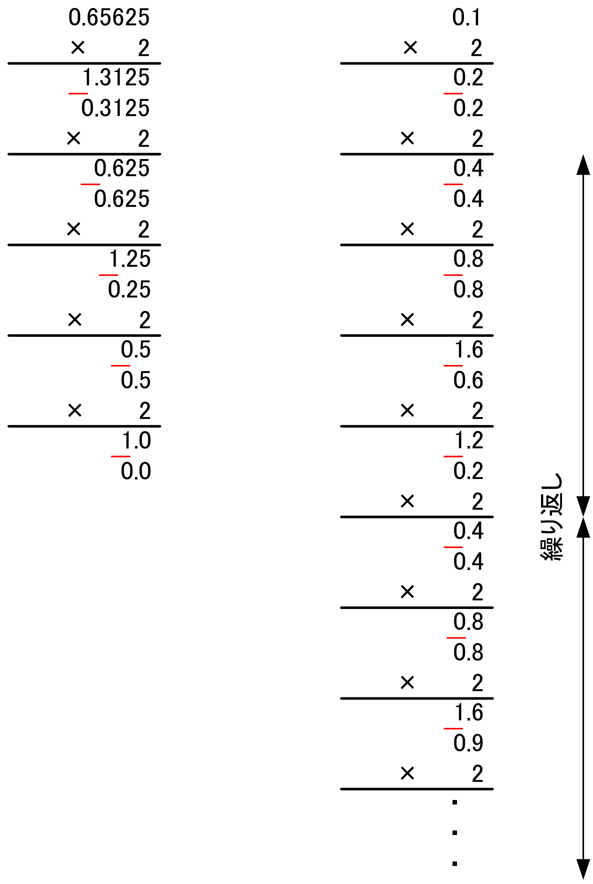

図 7: 小数の基数変換 (10 進数 →2 進数)