

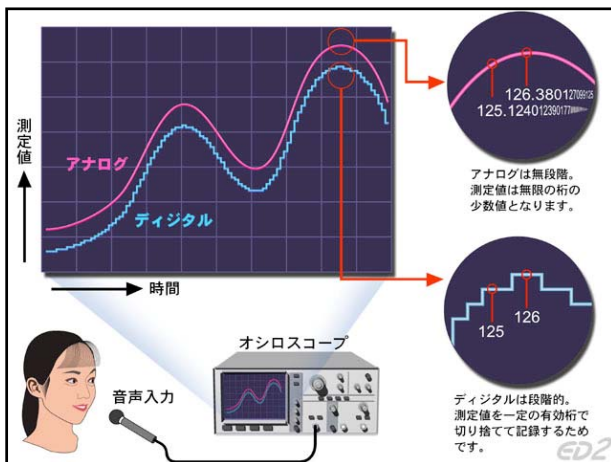
情報基礎

情報の符号化 2進数・16進数

Modified by Harumi Murakami
Originally written by Kota Abe

アナログとデジタル

- ・人が知覚できる自然界の情報はすべてアナログ
- ・アナログ (analog): 情報をなんらかの物理量(電圧等)で表現
 - ・連続量(どこまでも細かくできる)
- ・デジタル (digital): 情報を離散値で表現
 - ・離散量(とびとびの値)



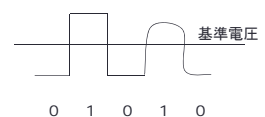
アナログとデジタル

- ・自然界の量をデジタル(数値)で厳密に表現しようとすれば、無限の桁数が必要
- ⇒「少数点以下3桁で四捨五入」あるいは「切り捨て」といったような丸めが行われる
- ・コンピュータでは全ての情報をデジタル化して扱う
 - ・内部では2進数を用いる

2進数・16進数

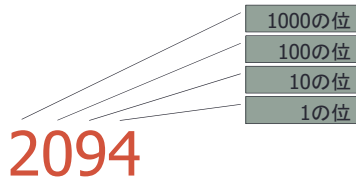
2進数

- ・各桁が 0 or 1
- ・0と1を電圧の高低で表せる
 - ・0V = 0, 5V = 1 など
- ・電気回路で扱いやすい
- ・ノイズに強い



10進数(復習)

- 各桁は0~9までの10通り
- 一桁上がると、桁の重みは10倍



$$2094 = 2 \cdot 1000 + 0 \cdot 100 + 9 \cdot 10 + 4 \cdot 1$$

2進数

- 各桁は0と1の2通り
- 2進数の1桁(0 or 1)をビット(Bit)と呼ぶ
 - Bit = Binary Digitの略
- 2進数 = Binary (バイナリ)
- 一桁上がると桁の重みは2倍

1010

$$1010 = 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 0 \cdot 1 = 10$$

8の位
4の位
2の位
1の位

基数の表記法

- 10進法以外の記法で数を表すと混乱する
- 基数(n進法のnのことを)右下に小さく書く

1010₍₁₀₎ — 10進法の1010

1010₍₂₎ — 2進法の1010
(10進法の10)

- 読むときはイチゼロイチゼロと読む
- センジュウと読んではいけない!

指で数える2進数

- 例) 指を曲げると1, 伸ばすと0(逆でもよい)



0 0 0 0 0

10進数と2進数の対応

埋めてみよう

10進数	2進数	10進数	2進数	10進数	2進数
0	00000	6	00110	12	01100
1	00001	7		13	
2	00010	8	01000	14	01110
3	00011	9	01001	15	01111
4	00100	10	01010	16	
5	00101	11	01011	17	10001

演習

- 以下の2進数を10進数に変換せよ

• $1000_{(2)} =$

• $11001_{(2)} =$

2進数の加算

$$\begin{array}{r} 1101 \\ +) 101 \\ \hline \end{array}$$

10進2進変換

• $11_{(10)}$ を2進数に変換してみよう

$$\begin{array}{r} 2) 11 \\ \hline 2) \underline{5} \dots 1 \\ \hline 2) \underline{2} \dots 1 \\ \hline 1 \dots 0 \end{array}$$

10進2進変換

- $11 = 8 + 2 + 1$ と分解してもよい
- 2の累乗の和で表す
- $8 + 2 + 1 = 2^3 + 2^1 + 2^0 = 1011_{(2)}$
- 8の位と2の位と1の位に「1」

演習

- 以下の10進数を2進数に変換せよ
- $30_{(10)} =$
- $63_{(10)} =$

nビットで表せる数

- 10進数n桁の組み合わせは 10^n 通り
- 2桁だと00~99の100通り
- 2進数n桁(nビット)の組み合わせは 2^n 通り
- 2桁だと00, 01, 10, 11の4通り
- nビットで $0 \sim 2^n - 1$ まで表せる

10進数の数え方	2進数の数え方	8ビットの2進数	ランプの点灯で示すと
0	0	00000000	●●●●●●●●
1	1	00000001	●●●●●●●●
2	10	00000010	●●●●●●●●
3	11	00000011	●●●●●●●●
4	100	00000100	●●●●●●●●
5	101	00000101	●●●●●●●●
6	110	00000110	●●●●●●●●
7	111	00000111	●●●●●●●●
8	1000	00001000	●●●●●●●●
9	1001	00001001	●●●●●●●●
10	1010	00001010	●●●●●●●●
11	1011	00001011	●●●●●●●●
12	1100	00001100	●●●●●●●●
⋮	⋮	⋮	⋮
99	01100011	00001111	●●●●●●●●
100	01100100	00010000	●●●●●●●●
101	01100101	00010001	●●●●●●●●
⋮	⋮	⋮	⋮
254	11111110	11111110	●●●●●●●●
255	11111111	11111111	●●●●●●●●
256	100000000		
257	100000001		
⋮	⋮		
⋮	⋮		

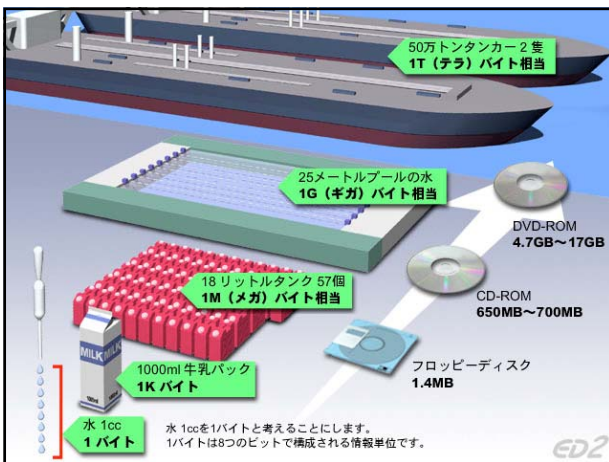
8ビットの2進数では11111111 (255)までしか数えられません。

ビットとバイト

- 1ビットという単位では細かすぎて使いにくい
- 8ビットをまとめて**1バイト (Byte)** として使う
 - 1バイト = 8ビット
- 1バイトで 256通り表せる
 - アルファベット(26 × 2=52文字)+ 数字(10文字)+ 各種記号を表すのに十分なサイズ
- **1bit = 1b, 1byte = 1B** と書くこともある
 - 1B = 8b
- コンピュータではデータをバイト単位で扱うことが多い

ビット, バイト, キロバイト...

- ビット (bit; b)
- バイト (byte; B) (=8ビット)
- **キロバイト (KB) (=1024バイト)**
- メガバイト (MB) (=1024キロバイト)
- ギガバイト (GB) (=1024メガバイト)
- テラバイト (TB) (=1024ギガバイト)
- ...



16進数(1)

- 2進数では桁数が多すぎて表記上不便
 - $50000_{(10)} = 1100001101010000_{(2)}$
 - **16進数**で表記するのが一般的
- 2進数を4桁ごとに区切り, それぞれを**16進数**1桁で表す
 - $1100|0011|0101|0000 \Rightarrow C350_{(16)}$
 - 16進数 = **Hexadecimal**
- 4ビットで 0~15 までの値を表現できる
 - 10~15 も1桁で表さないといけない
 - A~F を使う
- 表記法はいろいろ
 - $12AB_{(16)}$ $0x12AB$ など

16進数(2)

10進	2進	16進	10進	2進	16進
0	0000	0	8	1000	8
1	0001	1	9	1001	9
2	0010	2	10	1010	A
3	0011	3	11	1011	B
4	0100	4	12	1100	C
5	0101	5	13	1101	D
6	0110	6	14	1110	E
7	0111	7	15	1111	F

まとめ

- アナログとデジタル
- コンピュータでは全ての情報をデジタル化して扱う
 - 内部では2進数
- 2進数と10進数と16進数