

# 単位、情報量、デジタルデータ、CPU と高速化 ～ ICT 用語集 ～

水谷 正大

大東文化大学 mizutani@ic.daito.ac.jp

2014

## 大きさの表示法

国際単位系 (SI: Systèm International d'Unités)

大きさ	SI 接頭語	大きさ	SI 接頭語
$10^1$	da (デカ)	$10^{-1}$	d (デシ)
$10^2$	h (ヘクト)	$10^{-2}$	c (センチ)
$10^3$	k (キロ)	$10^{-3}$	m (ミリ)
$10^6$	M (メガ)	$10^{-6}$	$\mu$ (マイクロ)
$10^9$	G (ギガ)	$10^{-9}$	n (ナノ)
$10^{12}$	T (テラ)	$10^{-12}$	p (ピコ)
$10^{15}$	P (ペタ)	$10^{-15}$	f (フェムト)
$10^{18}$	E (エクサ)	$10^{-18}$	a (アト)
$10^{21}$	Z (ゼタ)	$10^{-21}$	z (zepto)
$10^{24}$	Y (ヨタ)	$10^{-24}$	y (ヨクト)

真空中の光速  $c = 2.998 \times 10^8$  [m/sec],

Avogadro 数  $N_A = 6.022 \times 10^{23}$  [mol<sup>-1</sup>], 1 光年  $9.46 \times 10^{15}$  [m], 太陽との距離  $au = 1.50 \times 10^{11}$  [m],

地球赤道半径  $6378.1$  [km], 電子半径  $2.818 \times 10^{-15}$  [m], 素電荷  $e = 1.602 \times 10^{-19}$  [C], 電子質量

$m_e = 9.109 \times 10^{-31}$  [kg], 微細配線で使われる Cu の直径  $2.5$  nm, 観測可能宇宙の原子数  $\sim 10^{80}$  個

## 記憶装置の容量

- **RAM** ( Random Access Memory ) 512MB ~ 4GB ( 記憶保持に電力必要、マイクロ化、超高速は高価 )
- **HDD** ( Hard Disk Drive ) 256GB ~ 5TB ( 安価、耐衝撃性低、かさばる )
- **SSD** ( Solid State Drive ) 64GB ~ 1TB ( 耐衝撃性高、高価、高速 )

### 生活用品の記憶装置

- ビデオレコーダ 500GB ~ 2TB
- CD 700MB、DVD 4.7GB、Blu-ray 25GB
- SD カード 1GB ~ 64GB、USB メモリ 1GB ~ 128GB
- iPhone 16GB ~ 64GB

## どれだけの情報量が必要か

- 4塩基からなる 32 億塩基対の DNA 情報は 800MB
- 人生の全体験の記録には 10TB ( Gordon Bell )  
全人類 80 億人分で 80 ゼタ B
- 地デジの 1 時間録画サイズ：標準 8GB 3 倍モード 4GB
- YouTube の 1 時間動画サイズ：0.6GB(HD) ~ 2GB(FHD)
- YouTube に 60 時間/1 分アップロード、1 日に 40 億回再生
- Google が処理するデータは 20 ペタ B/1 日 ( 1/2008 )
- NSA の新データセンタのディスク容量 5 ゼタ B ( 7/2013 )



## 2進数表示

- 10進数表現で使える記号は 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 の 10 個  
 $[123]_{10} = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$
- 2進数表現で使える記号は 0 と 1 の 2 個 (桁上がりに注意)

2進数表現  $[b_n b_{n-1} \dots b_1 b_0]_2, b_i \in \{0, 1\}$  の 10進表現

$$[b_n b_{n-1} \dots b_1 b_0]_2 = [b_n 2^n + b_{n-1} 2^{n-1} + \dots + b_1 2^1 + b_0 2^0]_{10}$$

### 2進数計算

$$\begin{array}{r} 0 \\ +) 0 \\ \hline 0 \end{array} \quad \begin{array}{r} 0 \\ +) 1 \\ \hline 1 \end{array} \quad \begin{array}{r} 1 \\ +) 0 \\ \hline 1 \end{array} \quad \begin{array}{r} 1 \\ +) 1 \\ \hline 10 \end{array}$$

0 と 1 の和の結果を同じ桁でみると XOR (排他的論理和) と同じ

$$\begin{array}{r} 0 \\ \times) 0 \\ \hline 0 \end{array} \quad \begin{array}{r} 0 \\ \times) 1 \\ \hline 0 \end{array} \quad \begin{array}{r} 1 \\ \times) 0 \\ \hline 0 \end{array} \quad \begin{array}{r} 1 \\ \times) 1 \\ \hline 1 \end{array}$$

## 2進計算の例

$[0]_2 + [0]_2 = [0]_2$	$[0]_2 \times [0]_2 = [0]_2$
$[1]_2 + [0]_2 = [1]_2$	$[1]_2 \times [0]_2 = [0]_2$
$[0]_2 + [1]_2 = [1]_2$	$[0]_2 \times [1]_2 = [0]_2$
$[1]_2 + [1]_2 = [10]_2$	$[1]_2 \times [1]_2 = [1]_2$

Table: 2進数の加法と乗法の基本

$$[11]_2 + [01]_2 = [100]_2 = [4]_{10}, [1001]_2 + [0110]_2 = [1111]_2 = [15]_{10}$$

$$\begin{array}{r}
 \phantom{\times)} \phantom{1} \phantom{1} \\
 \phantom{\times)} \phantom{1} \phantom{0} \\
 \hline
 \phantom{\times)} \phantom{0} \phantom{0} \\
 \phantom{\times)} \phantom{1} \phantom{1} \\
 \hline
 \phantom{\times)} \phantom{1} \phantom{1} \phantom{1}
 \end{array}
 \Rightarrow [11]_2 \times [10]_2 = [111]_2 = [6]_{10}$$

## 情報の単位 bit

歩行者信号から情報を得ることができる

$$\text{歩行者信号} = \begin{cases} \text{赤} & \rightarrow 1 \\ \text{青} & \rightarrow 0 \end{cases}$$

**1bit:** 1機の歩行者信号が伝える情報量 ([0] または [1] の2通り)

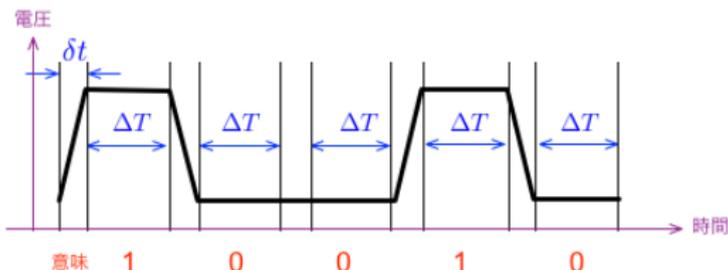
2機の歩行者信号の情報は 2 bits ([00],[01],[10],[11] の4通りを区別可能)

**$n$  bits** の情報  $\underbrace{[b_{n-1} \dots b_1 b_0]}_{n \text{ 個の } 0 \text{ または } 1} \Rightarrow 2^n \text{ 個の状態を区別可能}$

- Q1 A-Za-z の 52 文字を識別するためには、ローマ字 1 文字当たり何ビットの情報を有するか？
- Q2 漢字を含めた日本語 1 文字を識別するためには 1 文字当たり何ビットを割り当てればよいか？

## ビット列とその電気的実現

- **ビット列** (bit signal) とは、**2つの記号** (通常、0と1と称する) からなる記号列
- ビット列を、1は高い電圧を  $\Delta T$  時間維持、0は低い電圧を  $\Delta T$  時間維持して**電気的に実現可能**。 $\Delta T$  を小さくすれば**単位時間当たりのビット数**は多くなる



**Figure:** 5 ビット列 10010 を表す電気信号の時間変化。高い電圧と低い電圧間の遷移時間を  $\delta t$ 、電圧持続時間を  $\Delta T$  とすると、5 ビットを送信するために  $5(\Delta T + \delta t)$  時間が必要。

## デジタルデータはビット列で表現される

デジタルデータ ( 離散データ )  $\equiv d$  個の記号  $\{0, 1, \dots, d-1\}$  からなる記号列  $s$

$$s = s_1 s_2 \dots s_i \dots, \quad s_i \in \{0, 1, \dots, d-1\}$$

- 例) 任意の言語の文字列 ( ローマ字、ひらがな、漢字など )
- 任意の離散データはビット列で表すこと可能。
  - $2^n \geq d > 2^{n-1}$  を満たす  $n$  を求めて、 $n$  ビットで1つの離散記号を表すとすればよい。
- 例) ローマ字 ( A-Z, a-z の 52 文字 ) の 1 文字は、  
 $64 = 2^6 > 52 > 2^5 = 32$  に注意すると、6 ビット表現される

A	000000	a	011011
B	000001	b	011100
	⋮		⋮
Z	011010	z	110100

## bit より大きな情報の単位 Byte

- **8 bits=1 Byte** (バイト) を 1 つの単位とする情報単位  
B と略記。  $1KB = 2^{10} = 1024B$  などとすることも多い
- 0-1 のビット列を 8 個ずつに区切ると、1Byte は 2 進数表示で 8 桁  $[b_7b_6b_5b_4b_3b_2b_1b_0]_2$ ,  $b_i \in \{0, 1\}$  に相当  
 $[00000000]_2 = [0]_{10}$  から  $[11111111]_2 = [255]_{10}$  まで **256** 個の整数

Shift\_JIS 文字符号化では**全角文字 1 つを 2Byte** ( $2^{16} = 65536$  個を区別可能) で表す (ASCII 文字 (128 文字) 7bit で表示できる)。**UTF** 文字符号化は可変バイト長。

**IP パケットは最大 64KB** であるが、途中のネットワークのプロトコルや媒体ごとに固有の最大転送単位 (**MTU**) が決まっており、MTU を越えるときには IP パケットが分割されて送信される (**IP フラグメント化**)。

# デジタルコンピュータ

電磁気（光も含む）や機械動作などの物理現象を用いた装置を組み合わせ構成され、デジタルデータを計算（処理、表示、蓄積・呼び出し）する情報処理システム

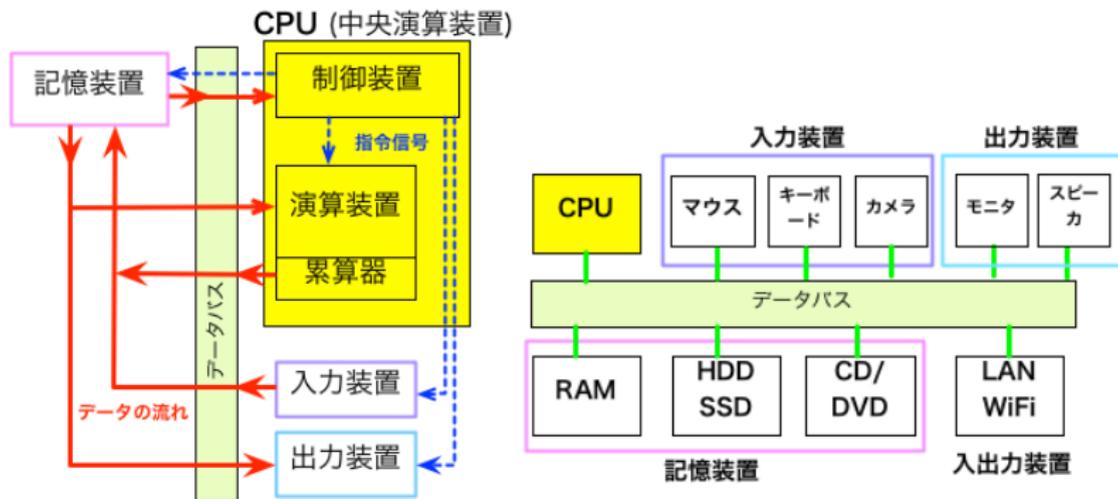


Figure: 制御装置から各装置に指令信号を送られている。制御装置が何をするかは規則装置内にあるプログラムに従う。データはデータバスを経由する。

## CPU での計算

- コンピュータ内の回路はそれぞれの装置ごとに一定の時間間隔（クロック）[sec] に合わせて処理
- 1秒間に刻むクロックの回数をクロック周波数（単位 Hz：ヘルツ）
- 1つの CPU 命令（instruction）には、少なくとも、メモリからフェッチ IF、デコード ID、実行 EX、メモリへのライトバック WB の4基本工程（各工程で最低1クロックは必要）
- 命令スループット（単位時間あたりに実行できる命令数）を向上させるために、命令レベルを並列化する命令パイプライン（各基本工程のための回路を遊ばせておかない）が使われている
- IPC=1クロックあたりに実行可能な命令数



## CPU 計算をしてみよう

「デジタル作法」 Kernighan, p47-59 <http://kernighan.com/toysim.html>

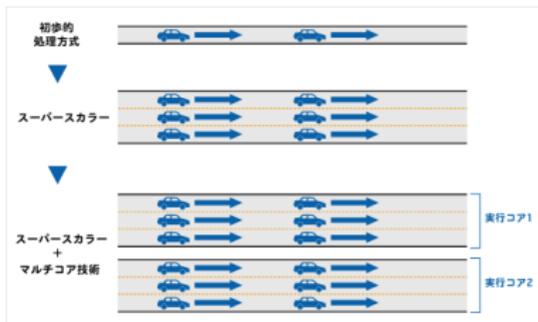
CPU は先頭メモリ番地から初めて、Fetch(取出し)でメモリから次の命令を読み込み、Decode(解読)でその命令が何をするかを判定し、Exec(実行)で命令を実行し、WriteBack(書込み)で結果をメモリに書き出すサイクルを繰り返す。CPU 内には計算した数値を保持するための**累積器**(accumulator)という記憶域がある。

### CPU命令

```
get      get a number from keyboard into accumulator
print   print contents of accumulator
load Val load accumulator with Val (Val unchanged)
store M  store contents of accumulator into memory location M (accumulator unchanged)
add Val  add Val to contents of accumulator (Val unchanged)
sub Val  subtract Val from contents of accumulator (Val unchanged)
goto L   go to instruction labeled L
ifpos L  go to instruction labeled L if accumulator is >= zero
ifzero L go to instruction labeled L if accumulator is zero
stop     stop running
Num      initialize this memory location to numeric value Num (once, before program runs)
```

## CPU 性能を向上させる要素

- クロック周波数を上げる
- スーパースカラー（回路を複数化）
- マルチコア（1CPU に複数の実行コアを搭載）
- アプリケーションの対応



[http://japan.intel.com/contents/museum/mpuworks/page\\_2.html](http://japan.intel.com/contents/museum/mpuworks/page_2.html)

## CPU の性能指標

**MIPS** (Million Instructions Per Second) = クロック周波数 × IPC ÷  $10^6$

**FLOPS** (Floating-point Operations Per Second)

**SPEC** (Standard Performance Evaluation Corporation) 公平で全体的ベンチマーク

## CPU vs メモリ

### CPU のフェッチは超高速

8 コアのプロセッサ intel Xeon E5 のクロック周波数は 3GHz。1 クロックに  $\frac{1}{3GHz} = 0.33 \times 10^{-9}$  [sec] つまり 0.33 ナノ秒しかかからない。この間、光は  $c \times \frac{1}{3} \times 10^{-9}$  [sec] =  $10^{-1}$  [m] つまり 10cm しか進まない！

### メインメモリからのフェッチは遅過ぎる

Xeon E5 に適合するメモリ DDR3 ECC メモリ動作はクロック周波数 200MHz で約  $\frac{2}{3} \times 1,866$  MHz で 8B (=64 ビット) の高速メモリ転送 (15[GB/sec] にも達する) を行うことができるが、メモリと CPU とのバス接続を考えると、メインメモリからフェッチするのために一般に 20 ~ 50 ナノ秒かかる。

より高速なメモリアクセスのために、直近でつかわれた情報を **キャッシュ** (cache) メモリとして CPU 内に格納しておき、メインメモリにアクセスする機会を減らして CPU の高速処理を実現する。CPU キャッシュは高速だがたいへん高価なメモリで、CPU コアに近い方から L1,L2,L3 キャッシュと呼ばれ、この順にアクセス速度が遅くなる (Xeon E5 では 25MB の L3 キャッシュを持つ)。

## Moore の法則 ~ 半導体の集積密度は2年で倍増

The complexity for minimum component costs has increased at a rate of roughly a factor of two per year. Certainly over the short term this rate can be expected to continue, if not to increase. Over the longer term, the rate of increase is a bit more uncertain, although there is no reason to believe it will not remain nearly constant for at least ten years. That means by 1975, the number of components per integrated circuit for minimum cost will be 65000.

I believe that such a large circuit can be built on a single wafer.

Cramming More Components onto Integrated Circuits

### Moore の法則の終焉?

現在 22nm 程度の**プロセスノード**（集積デバイスのゲート配線幅）の微細化の限界は7~5nmで限界を迎えるという（2020年頃）。原子数個分の幅になると量子効果が大きくなるためだ。理論的には**分子コンピュータ**、**量子コンピュータ**が新たなパラダイムとして実用化を目指した研究がなされている。