

コンピュータ概論(1), (2)

医学部医療情報学

平野 章二

講義の内容

- コンピュータ概論(1): 計算機の仕組み
 - 基礎用語
 - 計算機のハードウェア
 - 計算機のソフトウェア
 - まとめ
- コンピュータ概論(2): 計算機ネットワーク
 - 計算機ネットワークの基礎
 - インターネットの歴史
 - これからのネットワーク:
ブロードバンド通信, グリッドコンピューティングなど
 - まとめ

計算機のしくみ: 基礎用語

アナログとデジタル

- アナログ (analog)
 - ある量またはデータを, 連続して変化する物理量で表現すること(広辞苑)
 - 時間的または空間的に連続して変化する量をアナログ量といい, データがこのような量で表わされていることをアナログという(アスキーデジタル用語辞典)
- デジタル (digital)
 - ある量またはデータを, 有限桁の数字列(例えば二進数)として表現すること(広辞苑)
 - 有限桁の数値で表わされた量をデジタル量といい, データがこのような量で表わされていることをデジタルという(アスキーデジタル用語辞典)

ソフトウェアとハードウェア

2進数による表現

- ソフトウェア
 - コンピュータのプログラムを抽象的にとらえる呼称。コンピュータの運用に関する手順や処理する情報などを含めてもいう(広辞苑)
 - 情報を表現・伝達する媒体とは区別して、情報の内容を指す語。放送の番組や記録された音楽・映像など
- ハードウェア
 - コンピュータシステムで、トランジスタ・集積回路などから組み立てた計算機自体を、情報媒体に記録されたプログラム(ソフトウェア)と区別して呼ぶ語。放送のための設備、録音・録画のための装置などもいう(広辞苑)

- 2進数
 - 2を基数として数値を表現する記法 (cf. 10進数)
 - 計算機の内部では、off/onの2状態をそれぞれ0と1に割り当て、2進数で処理する

10進数	2進数	10進数	2進数	10進数	2進数
0	0000	5	0101	10	1010
1	0001	6	0110	11	1011
2	0010	7	0111		
3	0011	8	1000		
4	0100	9	1001		

$$\begin{aligned}(1011)_2 &= 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\ &= 1 \times 8 + 0 \times 4 + 1 \times 2 + 1 \times 1 \\ &= 11\end{aligned}$$

ビットとバイト

- ビット (bit)
 - ある情報を 0 と 1 の二種の記号の系列(二元符号)で表現するとき、その一つの記号をいう。例えば、二進数で情報を表現した時、そのある桁を表す 0 または 1 の数字
 - 情報量の単位。左か右か、0 か 1 か、など全く等確率な二者択一の質問に対する答の持つ情報量が 1 ビットに相当する
- バイト (byte)
 - 情報量の単位。コンピュータの内部で情報処理を行うとき、一単位として取り扱われる情報記号(ビット)のブロック
 - 通常、8 ビットを 1 バイトとする

ビットとバイト

- ビット数と表現可能な値の範囲
 - 0 から $2^n - 1$ まで
 - 8 ビット (1 バイト) ... 0000 0000 (0) から 1111 1111 (255) まで
 $2^8 = 256$ 通り
 - 16 ビット (2 バイト) ... $2^{16} = 65,536$ 通り
 - 32 ビット (4 バイト) ... $2^{32} = 4,294,967,296$ 通り

ビットによる情報表現

- 文字情報

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

例: 1 辺ドットの 'A' 1 文字
 $8 \times 8 = 64$ ビット (8 バイト)

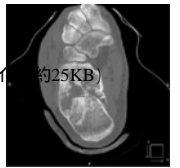
アルファベット等、頻用記号には
 8 ビット (1 バイト) の番号を対応付け
 例: A=65 (0100 0101)
 B=66 (0100 0110)
 必要なデータ量を削減

漢字は 16 ビット (2 バイト) で表現
 例: 水=(1011 1111 1110 0101)

身近な情報量

- 情報量の例

- 新聞 1 ページ (漢字約 13,000 字)
 - 16 ビット \times 13,000 文字 = 208,000 ビット (26,000 バイト、約 25 KB)
- 医用画像 (CT 画像, 512×512 画素)
 - 16 ビット (65536 階調) $\times 512 \times 512$
 = 4,194,304 ビット (524,288 バイト、約 512 KB)
 - (参考) フロッピーディスク 1 枚の容量 = 1.44 MB, CD 1 枚 = 650-700 MB)



- 情報量の補助単位

- 計算機は 2 進数なので、2 の乗数で表現
- K (キロ) = $2^{10} = 1,024$ 倍
- M (メガ) = $2^{20} = 1,048,576$ 倍
- G (ギガ) = $2^{30} = 1,073,741,824$ 倍

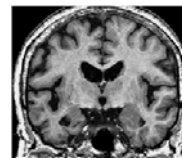
参考: SI 補助単位
 K = $10^3 = 1,000$
 M = $10^6 = 1,000,000$
 G = $10^9 = 1,000,000,000$

圧縮と復元 (展開, 解凍)

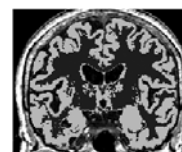
- 定められた手続きにより、余分な情報を取り除き、データ容量を低下させる
- 圧縮の例
 - AAAAAABBBBBAAA A6B5A3 (14 Bytes \rightarrow 6 Bytes)
- 圧縮法の種類
 - 可逆圧縮: 復元により元のデータが完全に再現される
 - LZW (GIF, TIFF), PNG など
 - 非可逆圧縮: 圧縮時にデータの損失を伴い、復元しても元のデータは完全には再現されない。
 - JPEG, MPEG など

圧縮と復元

- データの性質と圧縮法, 圧縮率



→ BMP 形式 (無圧縮): 149 KB
 GIF 圧縮: 26 KB
 JPEG 圧縮 (標準): 20 KB

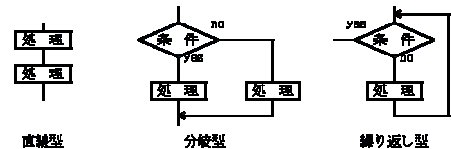


→ BMP 形式 (無圧縮): 149 KB
 GIF 圧縮: 14 KB
 JPEG 圧縮 (標準): 37 KB

計算機のしくみ: ハードウェア

計算機による情報処理

- 問題の解法をアルゴリズム(手続き)として記述
- アルゴリズムの構成要素:
 - 端子, 処理, 入出力, 条件判断, 分岐, 繰り返し, など
 - 各手続きの流れを図示したもの: フローチャート



- どのようにしてハードウェアで実現するか?

論理演算(ブール代数)

- 「真」=「1」, 「偽」=「0」
- 命題の真偽から結果の真偽を導く
 - 命題1: ユーザが実在する 「真(1)」
 - 命題2: パスワードが正しい 「真(1)」
 - 結果: ログオンを許可する 「真(1)」 → 1かつ1=1
- 論理積(AND): 全ての命題が真の場合のみ結果が真になる演算

	A(命題1)	B(命題2)	C(結果)
論理積の真理値表	1	1	1 ← 1×1=1
	1	0	0 ← 1×0=0
	0	1	0 ← 0×1=0
	0	0	0 ← 0×0=0

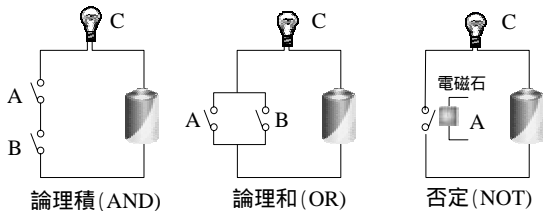
論理演算(ブール代数)

- 論理和(OR): 命題のいずれか1つでも真の場合に結果が真になる演算
- | | A(事柄1) | B(事柄2) | C(結果) |
|----------|--------|--------|-----------|
| 論理和の真理値表 | 1 | 1 | 1 ← 1+1=1 |
| | 1 | 0 | 1 ← 1+0=1 |
| | 0 | 1 | 1 ← 0+1=1 |
| | 0 | 0 | 0 ← 0+0=0 |
- 否定(NOT): 命題が真の場合は偽, 偽の場合は真になる演算

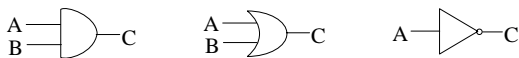
	A(事柄1)	C(結果)
否定の真理値表	1	0
	0	1

論理演算のスイッチによる実装

- 真 = スイッチON, 偽 = OFFとすると...



MIL記号



計算機の実現

- 論理演算回路の組み合わせで様々な機能を実現可能
- 例: 1ビット(1桁)の足し算

入力	結果	桁上り
0 0	0	0
0 1	1	0
1 0	1	0
1 1	0	1

0+0=0 1+0=1
0+1=1 1+1=10

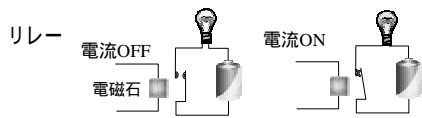
1ビット半加算器

処理の手続き(アルゴリズム)を論理演算で記述し, 対応する回路を組めば計算機を構築できる!

計算機の変遷

• 初期の計算機

- 手作業で配線を切替えて回路構成を選択
- リレーや真空管でON/OFFの状態を切替
 - 1938年「Z1シリーズ」(ドイツ) ... リレー
 - 1939年「ABC」(アメリカ) ... リレーと真空管
 - 1944年「Mark I」(アメリカ) ... リレーを使用した大型計算機
 - 1943年「COLOSSUS」(イギリス) ... 真空管
 - 1946年「ENIAC」(アメリカ) ... 真空管18,000本



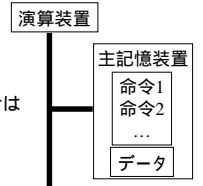
計算機の変遷

• フォン・ノイマン型コンピュータの登場

- プログラム内蔵方式: 処理手順(プログラム)を記憶装置(メインメモリ)に内蔵
- 逐次処理 命令を順番に読み取って実行
 - 1950年「EDVAC」(アメリカ)
 - 1949年「EDSAC」(イギリス)

- 現在のコンピュータのほとんどは フォン・ノイマン型

- 問題の解法を手続きとして記述できる場合は効率的に動作する
- それが出来ない問題は解が得られない



計算機の変遷

• 構成素子の变化

- リレーから真空管へ
 - 真空管: プレート等の電極を真空の管球に封入した素子。プレート間に流れる電流を電氣的に制御する。電球の応用。
 - リレーよりも高速動作が可能
 - 消費電力が大きく、小型化も困難
- 真空管からトランジスタへ
 - トランジスタ: 増幅機能をもつ半導体素子。スイッチとして利用可能
 - 半導体: 電気を通しやすい物質(導体)と通さない物質(絶縁体)の中間の性質を持つ物質。シリコンなど
 - 消費電力が非常に小さい、小型化が可能、長寿命



トランジスタの仕組み

• P型半導体とN型半導体

- 異なる電氣的性質をもつ粒子(P型は+ (正孔), N型は- (電子))をそれぞれ動きやすくしたもの
 - 正孔: 電子の抜け穴
- 半導体に微量の元素(硼素や砒素)を添加
- 3価の元素を加えるとP型に、5価の元素を加えるとN型に



P型半導体

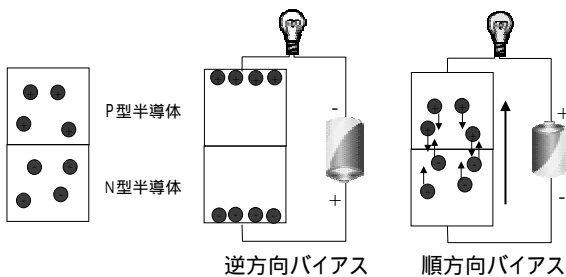


N型半導体

トランジスタの仕組み

• ダイオード

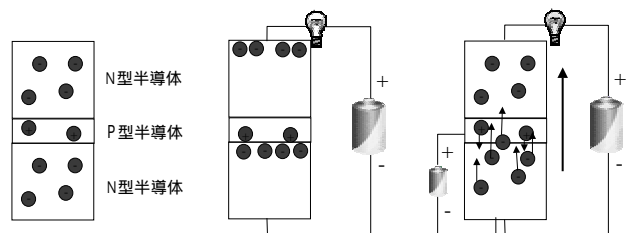
- P型半導体とN型半導体を接合(PN接合)した素子
- 整流作用をもつ



トランジスタの仕組み

• トランジスタ(NPN型)

- ごく薄いP型半導体をN型半導体で挟み込んで接合
- 電圧制御のスイッチとして動作可能

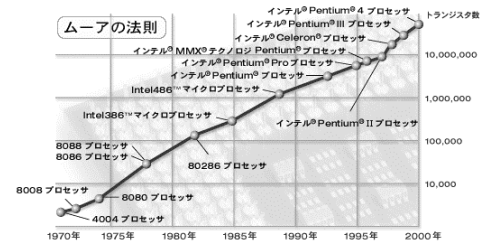


計算機の高機能化, 小型化

- トランジスタの発明により, 素子の小型化が進展
- 素子間配線の微細化により, 実装密度が向上
- 回路を構成する複数の素子を1つのチップにまとめた集積回路が出現 小型化, 高機能化
 - 集積回路 (Integrated Circuit; IC) 素子数1000程度
 - 大規模集積回路 (Large Scale Integration; LSI) 素子数数千から数万程度
 - 超大規模集積回路 (Very Large Scale Integration; VLSI) 素子数十万から百万程度
 - 超長大規模集積回路 (Ultra Large-scale Integration; ULSI) 素子数百万以上

集積化の例

- Intelプロセッサの集積密度

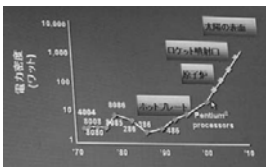


Source: Intel

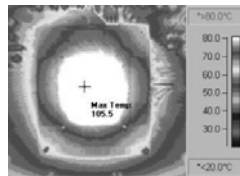
ムーアの法則: 半導体の集積密度は18~24ヶ月で倍増する (G.Moore, 1965)

集積化の課題

- 微細配線技術
 - 伝導抵抗の低減; アルミ配線から銅配線へ
 - プロセスレールの微細化; 現在は0.13 μm
- 発熱の抑制
 - 単位面積当たりの電力密度は現在でもホットプレート並, 数年後には原子炉レベルに?



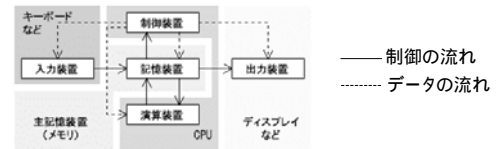
Source: ZDNet



Source: Transmeta

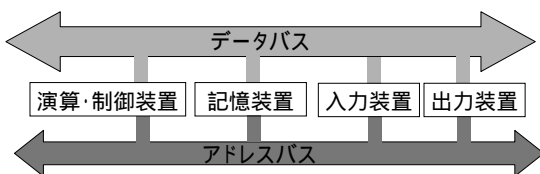
計算機の構成要素

- 計算機の5大機能: 入力, 記憶, 制御, 演算, 出力
 - 入力装置: 処理に必要な情報を入力する(キーボードなど)
 - 記憶装置: データやプログラムなどを記憶する(メモリ, HDDなど)
 - 制御装置: 各装置の動作を制御する(CPUが兼ねる場合が多い)
 - 演算装置: 計算を行う(Central Processing Unit; CPU)
 - 出力装置: 結果を出力する(ディスプレイなど)



計算機の構成要素

- 各装置はバスで繋がっている
 - データバス: データが流れる道
 - データバスが広い 一度に大量のデータをやり取りできる
 - アドレスバス: データの所在を示す道
 - アドレスバスが広い 巨大なデータ空間を扱える



クロック

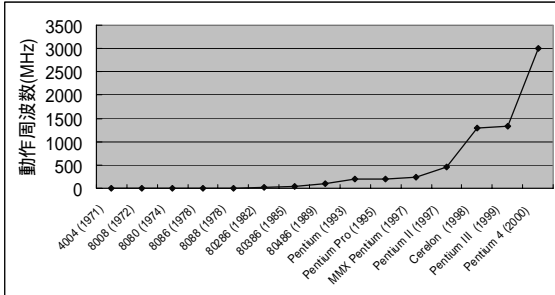
- 計算機の構成要素は膨大な数のスイッチ
- 全てのスイッチは同じタイミングで同期して動作
 - 処理の流れを統制し, 無駄を無くす
- クロック: 一定間隔で生成されるタイミングパルス信号



- クロック周波数: 一秒間に繰り返されるクロックの数
 - 1秒間に1回なら1Hz(ヘルツ), 100万回なら1MHz
- 外部クロック: 周辺回路のクロック
- 内部クロック: CPUやメモリなどが使用するクロック. 外部クロックを内部で数倍に高めて高速動作

処理速度の高速化

• クロック周波数の高速化



Source: Intel

主記憶装置

• メモリ(半導体メモリ)

- DRAM (Dynamic Random Access Memory)
 - 読み書き自由な記憶素子
 - 大容量で安価 (512MBで1万円程度)
 - 電源を切ると内容が消去される
- SRAM (Static Random Access Memory)
 - 読み書き自由な記憶素子
 - 電源を切っても内容は保存される
 - DRAMに比べて高速だが、高価
- ROM (Read Only Memory)
 - 原則的に読み込み専用メモリ
 - 特殊な方法(紫外線,あるいは電氣的)で書き込み,その後内容を保持
 - パソコン起動時の基礎処理プログラム(BIOS)の格納のほか,家電の制御装置などで広く利用

外部記憶装置

• ディスク装置

- フロッピーディスク
 - 磁気記録媒体の一つ。磁性体を塗布したプラスチックの円盤
 - 記録密度と大きさにより記録容量が決まる
 - 現在のPC/AT互換機では,3.5インチ,1.44MBが主流
 - 手軽で安価だが,読み書きが遅い
- ハードディスク
 - 磁性体を塗布したアルミニウムまたはガラスの円盤(通常複数枚)
 - 読み込みヘッドとディスクがごく僅か離れており,高速回転が可能で読み書きが高速。でも振動に弱い
 - 現在の主流は3.5インチ(デスクトップ用)または2.5インチ(ノート用)
 - 大容量で安価
 - 200GBで2万円弱

計算機の処理性能の推移

• 演算性能の推移

- 1977年 Cray スパコン Cray1: 0.15 GFLOPS
- 1983年 富士通スパコン VP200: 0.5 GFLOPS
- 1992年 日立スパコン HITAC: 8-32 GFLOPS
- 2002年 Intel Pentium 4 2.8GHz 5 GFLOPS
 - FLOPS: Floating-point Instructions Per Second;
 - 浮動小数点演算の処理速度
 - 現在のスーパーコンピュータの処理能力は数十Tera FLOPS
- 現在の20万円程度のデスクトップパソコンの処理能力は,一昔前のスーパーコンピュータに匹敵するほど高速

計算機のしくみ:ソフトウェア

ソフトウェアの重要性

- 計算機の能力を左右する
 - 計算機を「どのように動かすか」をソフトウェアが制御
 - 「ソフトが無ければただの箱」
 - ノイマン型コンピュータの出現により重要度が飛躍的に増加
- 人々の社会生活を变化させる
 - 基盤システムのソフトウェア化:
 - ライフライン制御,交通管制,金融,医療
 - ハードウェアからソフトウェアへ産業構造が变化
 - サービスの多様化,新産業の創出
 - 障害が社会生活に深刻な支障をもたらす
 - 複雑,高機能を実現すると同時に高い安定性と高速性が要求される

基本ソフトと応用ソフト

- 基本ソフトウェア (Operating System; OS)
 - キーボード入力や画面出力といった入出力機能やディスクやメモリの管理など、多くのアプリケーションソフトから共通して利用される基本的な機能を提供し、コンピュータシステム全体を管理するソフトウェア
 - ハードウェアの差を吸収
 - あるOS向けに開発されたソフトは違う機器でも基本的に動作する
 - ポピュラーなOS
 - Microsoft Windows (マイクロソフト)
 - Mac OS (アップルコンピュータ)
 - UNIX (AT&T ベル研究所), Linux
- 応用ソフトウェア (Application Software)
 - 文書作成、数値計算など、特定の目的のために設計されたソフト。
 - OS上で動作し、OSの機能に加えてユーザが必要とする機能を実装したもの。
 - Word, Excel, Internet Explorer など多数

ソフトウェアの開発

- 計算機が直接理解できるのは機械語のみ
 - 機械語: 演算装置がそのまま解釈、実行できる言語
 - 処理効率が高くコンパクトな記述が可能
 - プログラム構造の把握が困難であり、中規模以上のソフトウェアの開発には不向き
- 高級言語によりプログラムを作成し、機械語に変換して実行

高級言語 → (インタプリタ/コンパイラ) → 機械語

 - 高級言語: 処理を人間の言葉に近い表現で記述するための言語
 - コンパイラ: 高級言語で記述されたプログラムの内容を解釈し、機械語に変換するソフト。一通り変換を終えた後でプログラムを出力して実行するため、高速に動作する。
 - インタプリタ: プログラムの内容を逐次解釈しながら実行していくソフト

プログラムの例

- $6 + 2$
 - 機械語: 25 6 2
 - アセンブラ: ADD 6 2;
 - C言語他: 6+2
- AとBを足して、Cに代入する
 - アセンブラ: MOV EAX, A;
MOV EBX, B;
ADD EAX, EBX ;
 - C言語他: C=A+B;
- C言語プログラムの例

ソフトウェア開発言語

- 高級言語の利点
 - ソフトウェアの生産性・管理性・移植性が向上
- 高級言語の例
 - FORTRAN: 科学技術計算用。関数が豊富。
 - COBOL: 事務処理計算用。英文に近い記述が可能。
 - BASIC: インタプリタ言語。学習が容易。
 - C言語, C++: システム開発向き。構造化、オブジェクト指向プログラミングをサポートすると同時に、より機械語に近い操作も可能。
 - JAVA: C言語の利点を継承しつつ、移植性を大幅に高めた言語。OS上に仮想マシンを構築し、その上でプログラムを実行するため、OSや機器構成に依存しない汎用性の高いプログラムを作成可能。インタプリタ型言語。

計算機の仕組み:まとめ

- 計算機のハードウェア
 - 構成素子の集積化による小型化, 高性能化
 - 価格対性能比の飛躍的向上
 - 新たな発想に基づく超高速次世代計算機の開発
 - 光コンピュータ
 - 量子コンピュータ
- 計算機のソフトウェア
 - ノイマン型コンピュータの核
 - 高級言語の導入による開発の効率化, 高機能化
 - 社会システムへの浸透に伴い、プログラムの記述ミス(バグ)による不具合の影響が深刻化