



# コンピュータシステム (アーキテクチャ 第1回)



工学部 情報エレクトロニクス学科

大学院 情報科学研究所 情報理工学部門

堀山 貴史

# コンピュータシステム 今年度講義予定

情報理工学コース 2年秋ターム 専門科目（必修）

- ・ アーキテクチャ  
（担当：堀山 貴史 教授）
- ・ オペレーティングシステム  
（担当：堀山 貴史 教授）

# 講義資料について

- ・ アーキテクチャの回では教科書は指定しません。
- ・ 講義資料は、下記の URL からダウンロードしてください。
  - 毎回の講義スライドを置いておきます。

[https://art.ist.hokudai.ac.jp/~horiyama/comp\\_sys/](https://art.ist.hokudai.ac.jp/~horiyama/comp_sys/)

「コンピュータシステム」ELMS/Moodle のページからリンクしています

- ・ 参考書
  - 雨宮真人, 田中譲:「コンピュータアーキテクチャ」, オーム社  
(旧コースの教科書。昭和時代の本。絶版で入手困難)
  - パターソン & ヘネシー:「コンピュータの構成と設計 第5版(上・下)」,  
日経BP社 (名著だが、上下巻合わせて約9000円する。高価)

# 講義の進め方と評価

- ・ アーキテクチャの回では、毎回、レポート用紙を配布します。
  - － レポート用紙には、毎回、小問を出しますので、解答してください。  
また、質問や感想を書く欄を設けます。
  - － 授業中に時間を取って記入してもらい、授業の終わりに回収します。
- ・ 期末試験は、アーキテクチャ50点、オペレーティングシステム50点とし、原則として合計60点以上を合格とします。
  - － 基本的には、期末試験の得点で成績評価しますが、ボーダーラインぎりぎりの人については、レポートへの取り組み方も考慮して判断します。
  - － (参考)教育評価の3要素:「知識」「技能」「態度」

# 情報理工学演習Iとの関係

- ・ 水曜4限目の情報理工学演習I(必修)と連動して講義を進めます。
  - 情報理工学演習Iでは、  
秋タームに「コンピュータシステム」  
冬タームに「ネットワークとクラウド」  
の講義に関する演習を行います。
  - 「コンピュータシステム」の演習は7回分、そのうち計算機アーキテクチャに関する演習は4回分を予定しています。  
(詳しい日程は演習の時間に説明します)
  - 演習科目は出席して演習を行うことが重要なので、  
すべての演習課題に取り組み、レポートを提出することが  
演習の単位取得の必要条件です。

# アーキテクチャ講義内容(予定)

- 第1回 (2023.10. 2) 計算機アーキテクチャとは
- 第2回 (2023.10.11) 機械語命令と内部動作
- 第3回 (2023.10.18) アーキテクチャの基本知識(1)  
(分類と概観・初期のメインフレーム)
- 第4回 (2023.10.23) アーキテクチャの基本知識(2)  
(メインフレームの発展)
- 第5回 (2023.11. 1) アーキテクチャの基本知識(3)  
(マイクロプロセッサ以降)
- 第6回 (2023.11. 6) 並列処理アーキテクチャ
- 第7回 (2023.11.15) メモリアーキテクチャ

# 今回(アーキテクチャ第1回)の内容

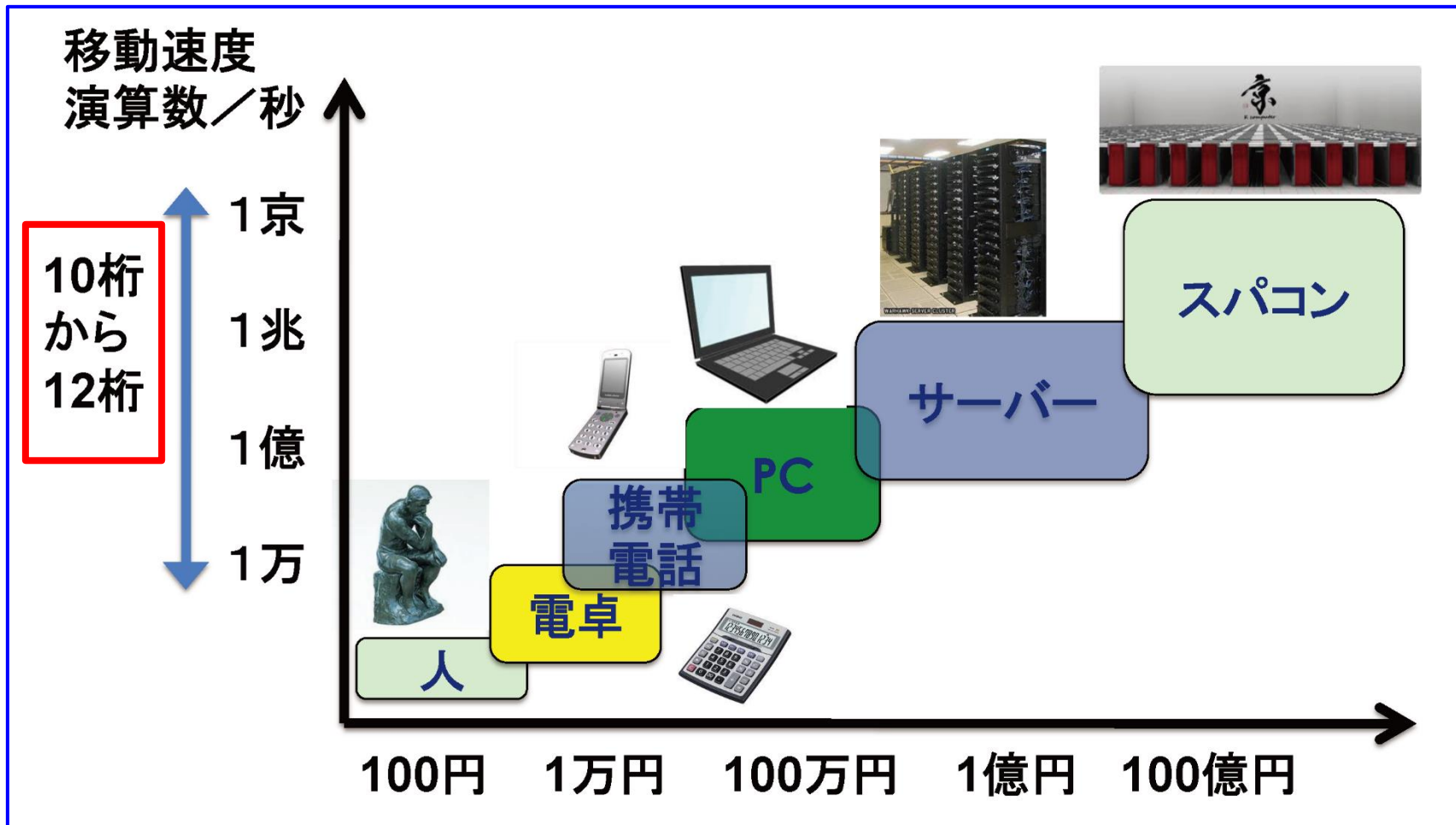
## 計算機アーキテクチャとは

- 現代社会におけるコンピュータシステム
- 計算機の利用形態の分類とその特性
- アーキテクチャの評価尺度とトレードオフ
- ハードウェアの構成要素
  - レジスタ、演算器、プログラムカウンタ、命令デコーダ、メモリ(主記憶)、バス、ディスク(2次記憶)、入出力装置
- ソフトウェアの構成要素
  - 機械語、ファームウェア、オペレーティングシステム、コンパイラ/インタプリタ、ソフトウェアライブラリ、アプリケーションプログラム

# 現代社会におけるコンピュータシステム

- 20世紀後半～21世紀初頭にかけて、コンピュータシステムはすさまじい技術革新を遂げた

(JSTさがけ「社会情報基盤」(安浦寛人代表) 資料より)

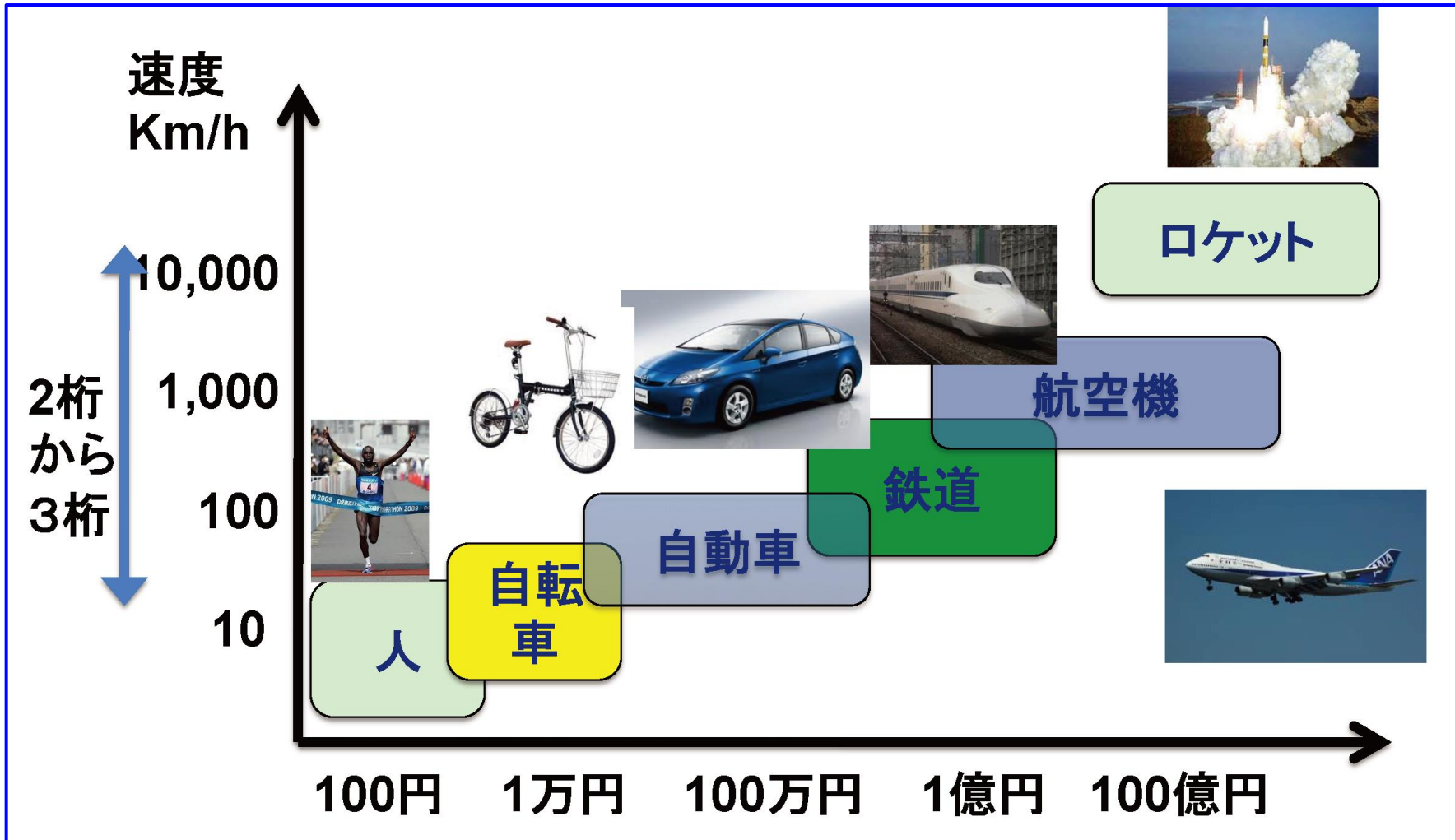




# (参考) 物理的な移動手段との比較

- ・ 宇宙ロケットでも高々100倍から1000倍の向上
  - コンピュータシステムの発展は桁違いにすさまじい。

(JSTさきがけ「社会情報基盤」 (安浦寛人代表) 資料より)



# 現代社会でのコンピュータの利用例

- ・ 車載コンピュータ
  - － エンジン制御、燃費改善、排ガス抑制、ABS、エアバッグ制御、ドライブレコーダー、カーナビ、自動運転
- ・ 携帯電話・スマートフォン・タブレット端末
  - － デジタルカメラ、音楽・ビデオ再生、ビデオ通話、電子メール、スケジュール管理、電話帳、地図、GPS、辞書、時計、電子マネー決済、電子チケット、電子ブック、バーコード、万歩計、健康管理
- ・ ヒトゲノム解析
  - － 1990年代の国際プロジェクトで1人分の解析に約10年かかった
  - － 今では、費用1000ドルで、1日で解析(個人で注文可能に)
- ・ WWWと検索エンジン
  - － 今では、新聞・出版、ラジオ・テレビ放送に代わる社会情報基盤に

# 計算機の利用形態の分類とその特性

- ・ デスクトップコンピュータ(desktop computer)
  - 個人が安価に小規模な処理を行うことが目的。PC (personal computer)とも呼ばれる。ディスプレイ、キーボード、マウスなどを備える。備え付けまたは別途入手したソフトウェアを使用。近年は、ラップトップ、ノート、タブレットと進化。
- ・ サーバ(server)
  - 複数ユーザに特定の処理を提供。多くの場合、ネットワークを通じて同時多数が利用。性能と信頼性が重要。サーバ用計算機の性能・価格は、用途に応じて、PC程度からスーパーコンピュータ(super computer)まで幅広い。Webサーバ、計算サーバ、データベースサーバなど。巨大なデータセンター(data center)を構築する場合もある。
- ・ 組み込みコンピュータ(embedded computer)
  - 専用プログラムまたはプログラム群を実行するために、製品に組み込まれて使われる計算機。家電、携帯、自動車、航空機など多種多様。普段見えないが台数は圧倒的に多い。必要最小限の性能で良いがコスト・電力の制約は厳しい。用途によっては信頼性も重要。

# 計算機アーキテクチャの役割

- ・ 計算機アーキテクチャ(computer architecture)の専門家は「アーキテクト」(architect = 建築家)と呼ばれる
- ・ 用途に応じて最適な計算機を設計することが仕事
  - ハードウェアとソフトウェアの分担をどうするか
  - ハードウェアの要素部品をどのように配置・配線するか
  - どのような手順で製造するか、検査や運用をどう計画するか
- ・ 設計が良いか悪いかを判断するためには「評価尺度」が必要
  - 数値化できる尺度があれば評価しやすい
  - ただし数値化しにくい要求事項もある

# 計算機アーキテクチャに対する要求

- ・ より多様な(applicability)
- ・ より大規模な問題を(capacity)
- ・ より高速に(speed)
- ・ より高い信頼度で処理する(reliability)
- ・ より使いやすい(usability)
- ・ より低価格な(cost-effective)
- ・ より省電力・低発熱な(low-power)

近年、重要性が増大

計算機が望まれる。

**どの2つを取っても互いに競合する(トレードオフの関係)**

－ (例) 速度と価格、使いやすさと速度、規模と価格

- ・ どの項目も際限のない要求であることが多い。

**良い妥協点(trade-off point)を見つけることが設計者の役割**

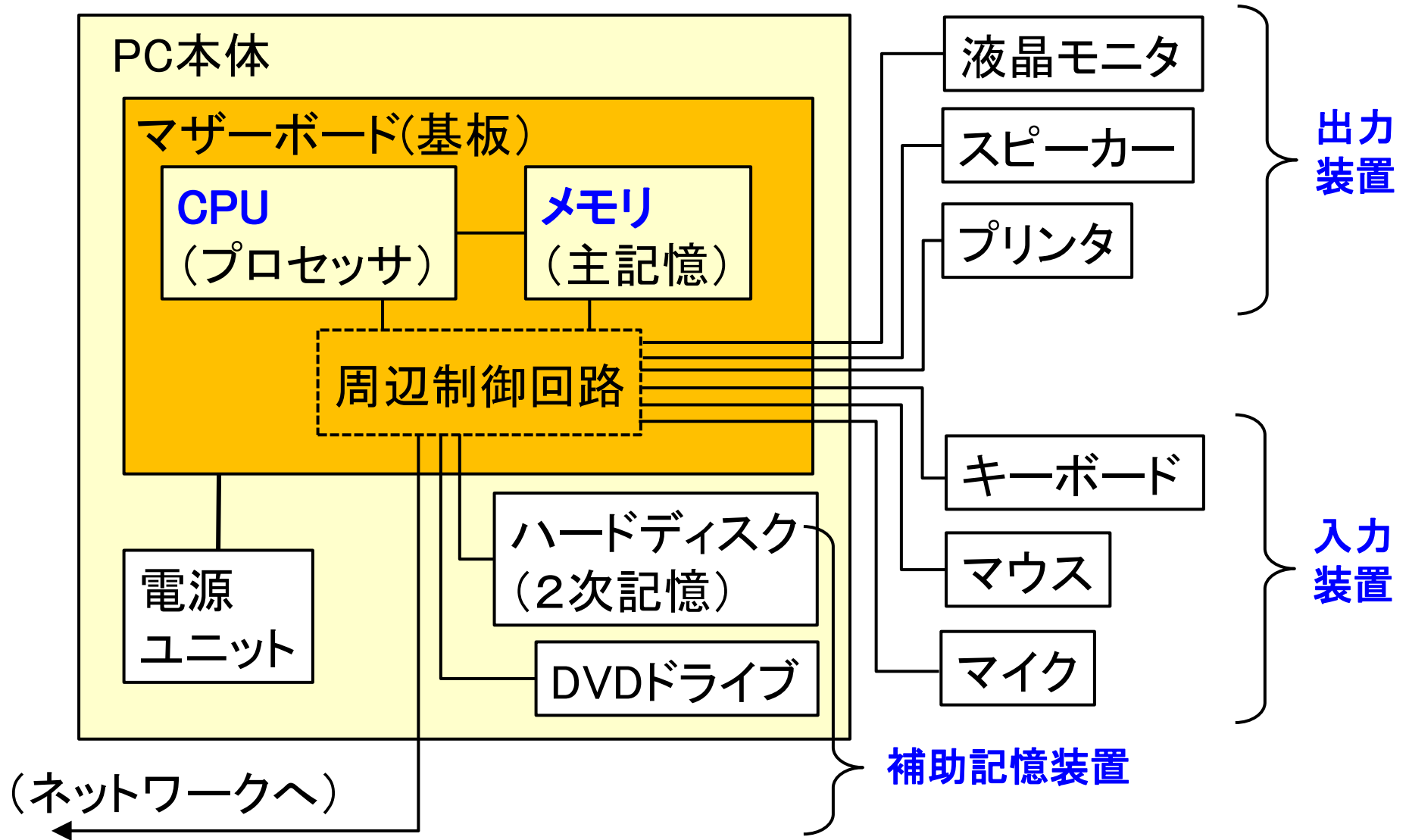
# 計算機アーキテクチャの評価尺度

- ・ 問題規模、計算速度、信頼度、価格、消費電力の項目は、数値化された尺度を持つので比較しやすい
  - 問題規模は記憶領域のサイズで表すことが多い  
Byte, kB, MB, GB, TB, ...
  - 高速性は1秒間に実行できる命令回数や演算回数で評価される  
MIPS(Mega Instructions Per Second),  
MFLOPS(Mega FLoating-point Operations Per Second),  
GFLOPS(Giga ...)
  - 信頼性は、誤動作から次の誤動作までの平均時間で評価される  
MTBF(Mean Time Between Failures)
  - 価格は円やドルで評価。消費電力はW(または電力量Wh)で評価
- ・ 多様性と使いやすさは数値化が困難
  - 評価困難だが重要。計算機やOSの栄枯盛衰に大きく影響している

# 計算機アーキテクチャの設計とは

- ハードウェアの基本構成要素を、どのように並べて、どのように配線するかを決める
- 機械語の命令セット(instruction set)を決める
  - 機械語命令は、データをどこからどこに移す、演算する、条件分岐する、などの基本的な操作を行う単位
  - 命令セットは、ハードウェアとソフトウェアのインタフェースとなる
- ハードウェア構成と命令セットは、互いに影響し合うので、より優れた設計のためには、両方同時に考えた方が良い
  - 機械語命令を高速に実行するためのハードウェア構成の工夫
  - ハードウェア部品の性能を引き出すための機械語命令の追加
  - ハードウェア構成と命令セットの一方がすでに固定されていて、それに合わせてもう一方を設計することもよくある

# 典型的なPCの内部構成



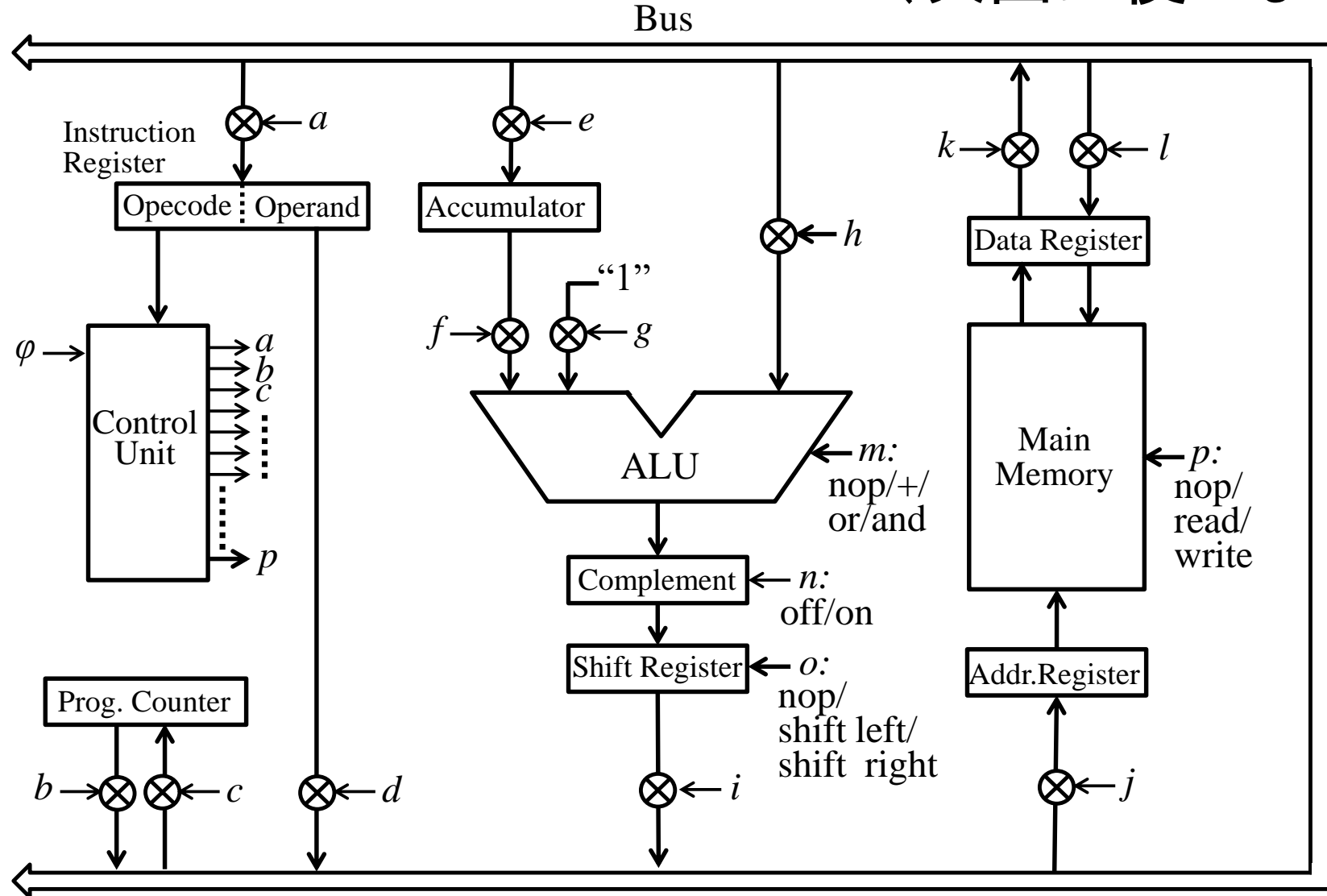


# 一般的なハードウェアの構成要素

- CPU(中央処理装置)(Central Processing Unit)
  - レジスタ (register)
  - 算術演算装置 (ALU: Arithmetic Logic Unit)
  - プログラムカウンタ (program counter)
  - 命令レジスタとデコーダ (instruction register & decoder)
- クロック発生器 (clock generator)
- メモリ(主記憶)(main memory)
- バス (bus)
- ハードディスク (2次記憶)(secondary memory)
- 入出力装置 (I/O device)
- 電源装置 (power unit)

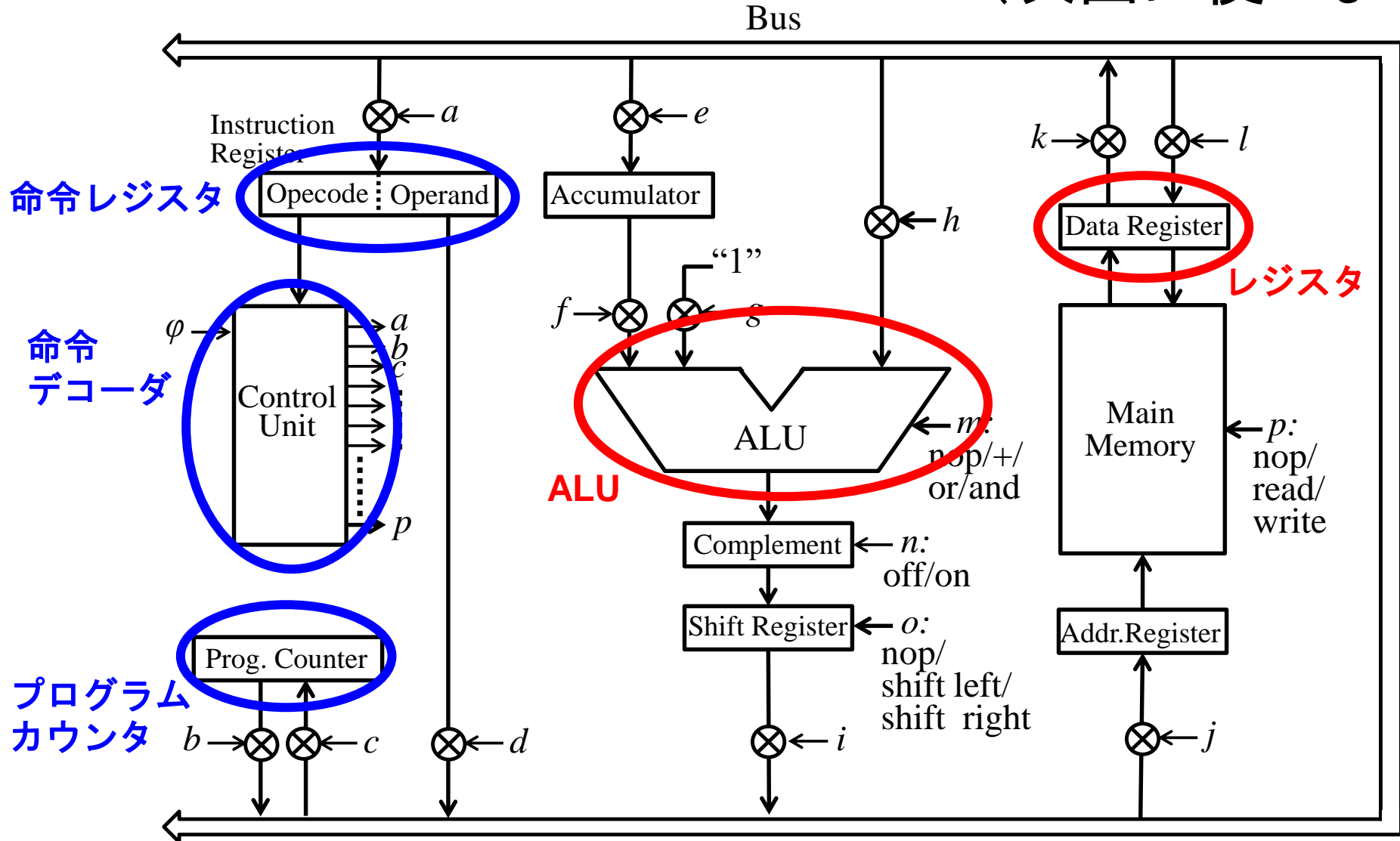
# 参考：基本的なアーキテクチャの構成

## (次回に使います)



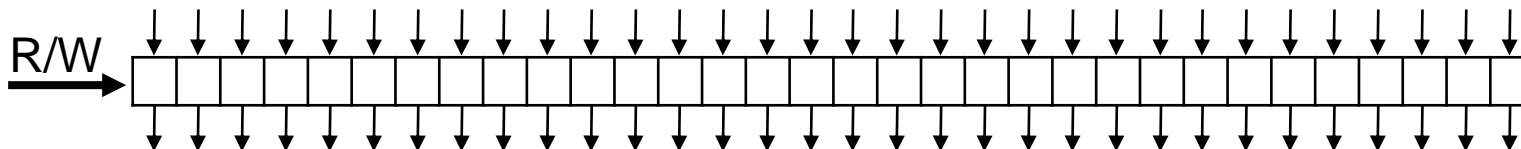
# 参考：基本的なアーキテクチャの構成

(次回に使います)

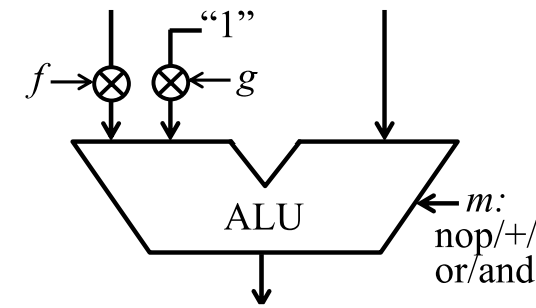


# レジスタ

- 1ワード(多くの場合32ビットまたは64ビット)の2進数を一時的に記憶する場所。
  - 通常は、1ワード=レジスタのビット長
  - CPUの内部に数個～数十個用意されている。
- 1回のクロック(同期信号)で書き込み、読み出しができる。
  - 1回書き込めば、電源を切らない限り何度でも読み出せる。
  - 新しいデータを上書きすると元のデータは失われる。
- デジタル電子回路で構成されている。
  - フリップフロップ回路を32個(または64個)並列に並べている。
  - 共通のRead/Write信号で、すべてのビットを並列に読み書きする。

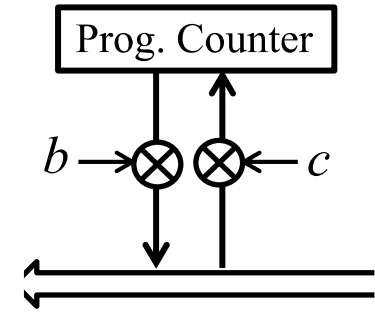


# 算術演算装置 (ALU)



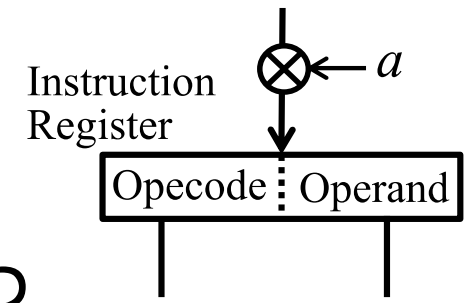
- 2進数の加減乗除の算術演算を行う装置
  - 多くの場合32ビットまたは64ビットの2進数を扱う
  - 小規模CPUでは、加算、正負反転、シフト(2倍、半分)しかできないものもある
- 高機能なCPUでは、高度な浮動小数点演算のALUを持つ
  - 科学技術計算では浮動小数点演算が重要
  - グラフィック計算や暗号処理に必要な積和演算を一気に実行できる演算器を持つプロセッサもある
- 簡単な演算は1クロックで実行できるが、複雑な演算では数クロック必要になることがある。他の装置は結果が出るまで(ALU回路の出力電圧が安定する時間まで)待たされる
- 高性能な算術演算回路の設計は、計算機アーキテクチャの中でも、非常に重要な技術

# プログラム カウンタ



- CPUの中で特別な意味を持つレジスタの1つ
- 現在実行中の機械語命令が格納されているメモリ番地を記憶している
  - 1命令実行するたびに1増やす
  - 分岐命令でジャンプするときは、行先のメモリ番地をプログラムカウンタに上書きする

# 命令レジスタ



- CPUの中で特別な意味を持つレジスタの1つ
- プログラムカウンタに書かれたメモリ番地から、機械語命令のデータを読み取って、一時的に記憶する場所
  - 機械語命令は1ワード(または数ワード)の2進数で書かれている
- 命令レジスタの各ビットの出力は、「**命令デコーダ**」につながっている
  - 命令デコーダは、命令レジスタの各ビットの0/1パターンに応じて、レジスタや演算器を正しく動作させる制御信号を発生させる
  - 具体的には、AND/OR/NOTなどの論理ゲート回路や、読み出し専用のメモリ回路などで作られている

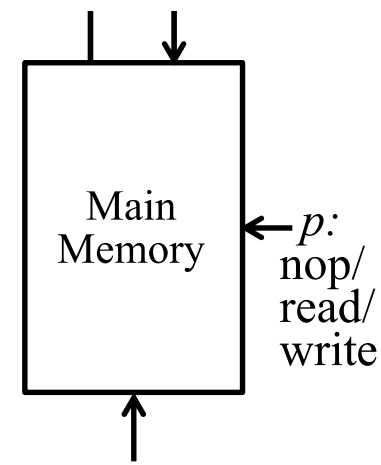
# クロック発生器

- 多くの計算機では、周期的な信号(クロックと呼ぶ)に同期して、すべての構成要素が一斉にデータのやり取りを行う
  - 各要素部品の動作時間がクロック周期に間に合うならば、クロック周波数が高いほど、計算機は高速になる
- CPUのクロック周波数が例えば2GHzの場合、毎秒20億回のクロック信号が、CPU内部の全てのハードウェア構成要素に供給されている
  - CPU内部にはクロック信号線が神経のように張り巡らされている
  - GHzクラスの高い周波数になると、電磁波になって逃げて行ったり、電線の間で自然発生するコンデンサに流れたりして、電力消費と発熱の問題が深刻になっている
  - 近年ではこれ以上クロック周波数を上げるのは限界に近い
  - 使っていない部分のクロック信号を、部分的に止めたりする工夫が行われている



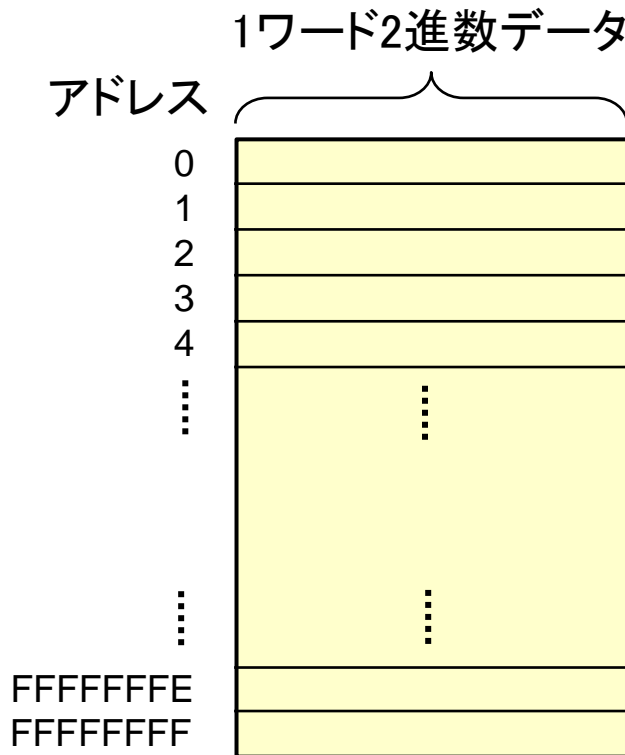
# メモリ（主記憶）

- プログラムとデータを格納する場所
- 0番地から始まる連続する番地（アドレス）を持ち、各番地に1ワードの2進数を格納できる
  - 現在では大容量の半導体メモリが使用されている
  - 最近では32～64GBのメモリ容量を持つノートPCも市販されている  
64GB = 約80億ワードのデータ容量
- 読み書きしたい番地を指定すると、その番地だけを直接アクセスできる（ランダムアクセス）
  - 磁気テープのように順番に読み書き（シーケンシャルアクセス）する必要はない。どの番地もほぼ同じ時間でアクセスできる
- ただし大容量のメモリは、アクセス速度が遅い
  - アドレスを指定して1回の読み書きをするのに、数クロック～数十クロックかかるので、その間、他の装置は待たされる

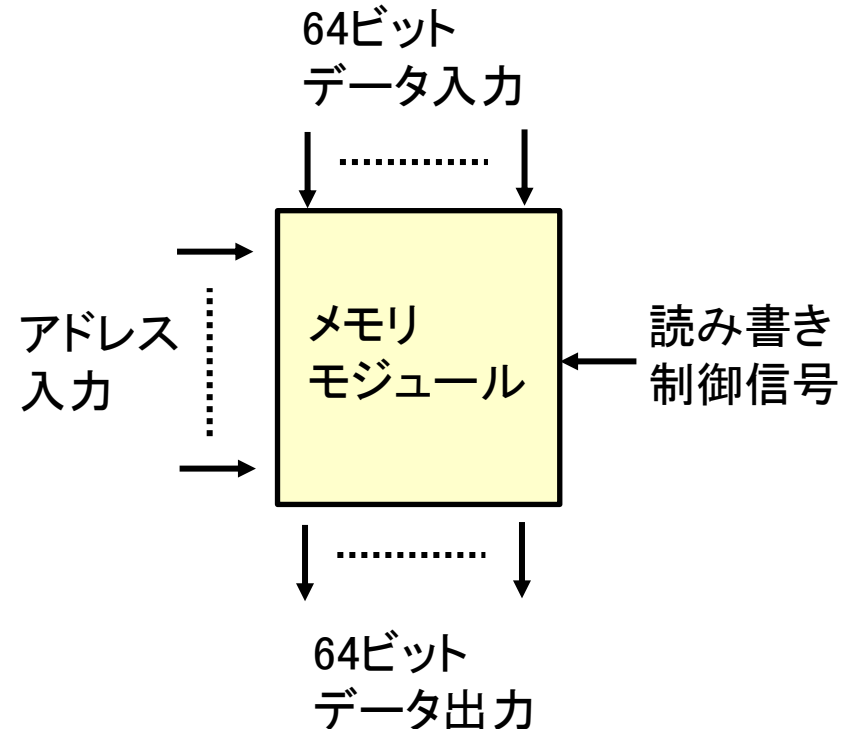


# メモリの構造

## 論理的構造



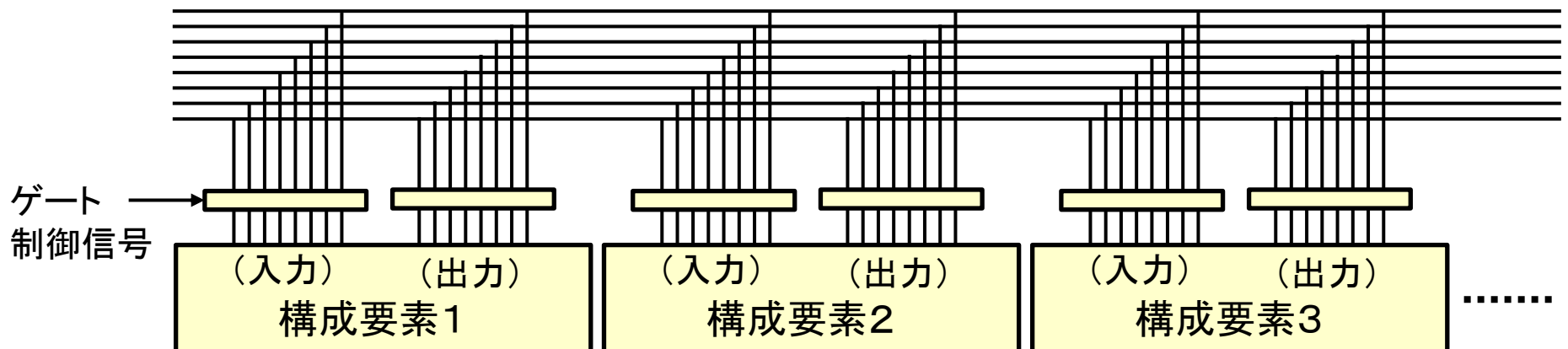
## ハードウェア要素としての構造



- アドレスとデータと書き込み信号を入力して一定時間待つと指定番地に書き込まれる
- アドレスと読み出し信号を入力して一定時間待つと、指定番地のデータが出力される

# バス配線方式

- ・ 計算機ハードウェアでは、ほぼすべての構成要素（レジスタ、演算器等）の間にデータを移動する可能性がある
  - 32本（または64本）の電線をあらゆる要素間に配線する必要がある
  - $n$ 個の構成要素があれば、ビット幅  $\times n(n-1)$  組の配線が必要  
→ 配線だらけになって、たいへん
- ・ すべての要素の入出力を共通に接続して、データ移動のタイミングを一括制御すれば、配線を1組に節約できる  
→ バス配線方式と呼ばれ、広く使われている

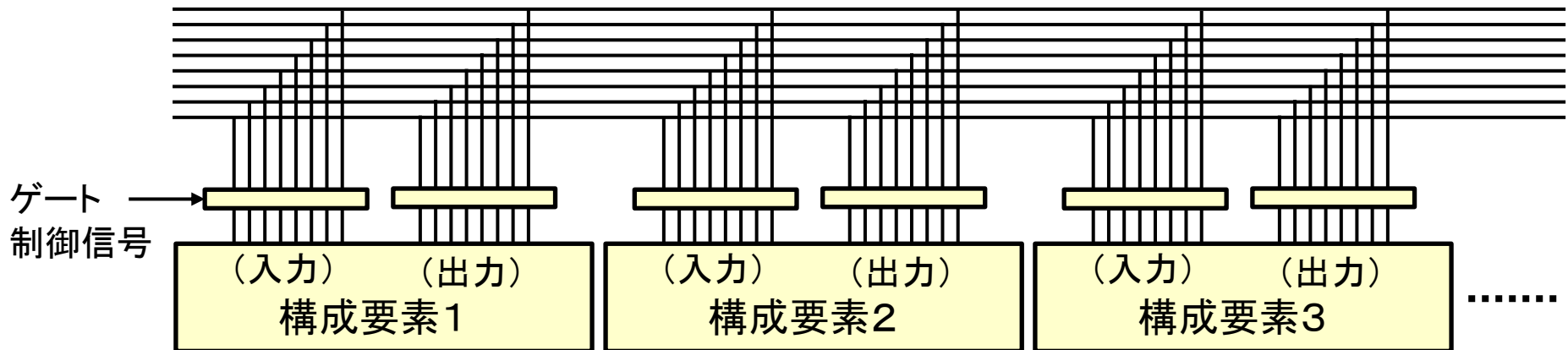


# バス配線方式

- ・ 計算機ハードウェアでは、ほぼすべての構成要素(レジスタ、演算器等)の間でデータを移動する可能性がある
  - 32本(または64本)の電線をあらゆる要素間に配線する必要がある
  - $n$ 個の構成要素があれば、ビット幅  $\times n(n-1)$  組の配線が必要  
→ 配線だらけになって、たいへん
- ・ すべての要素の入出力を共通に接続して、データ移動のタイミングを一括制御すれば、配線を1組に節約できる  
→ バス配線方式と呼ばれ、広く使われている
  - 2個以上の要素部品が同時に出力モードになると、過電流が流れる(最悪、焼損する)ので、そうならないよう制御する必要がある
  - 1個のバスでは複数ペアのデータ移動は同時にはできない
  - 複数のバスを用意すれば並行してデータ移動できるが、配線量が増える

# バスとゲート

- ・ ゲート: 制御信号によって、データ線をつないだり切り離したりするスイッチ
  - ゲートを開く/閉じる(open/close): データ信号を通す/通さない
  - 原則として、どこか1か所の出力と1か所の入力だけを開く



# ハードディスク（2次記憶、補助記憶）

- ・ 主記憶は電源を切るとデータが失われる（揮発性）
- ・ 高速ランダムアクセスできる半導体メモリは高価
- ・ 安価で不揮発性の記憶装置を計算機外部で使用する（2次記憶と呼ぶ）
  - － 2015年の時点では、ハードディスク（HDD; Hard Disk Drive）が最もビット単価が安い（2TBで約1万円）
  - － 磁気ヘッドで機械的に読み書きするので、ランダムアクセスが苦手
  - － 連続領域を順次アクセスするのは比較的高速
  - － 近年、フラッシュメモリを用いたSSD（Solid State Drive）の容量が急速に拡大している。HDDよりは高速だが、今のところランダムアクセスは主記憶よりかなり遅い。読み出しに比べて書き込みが遅い
- ・ 計算の実行は、主記憶の上で行う。2次記憶は、大量のプログラムやデータを長期保存するのに使われる

# 入出力装置

- ・ キーボード、マウス、ディスプレイ、プリンタ、マイク、スピーカ、ネットワーク機器など
- ・ CPUから見ると、2次記憶と同様のインタフェースで接続されることが多い
  - 主記憶の特定の領域がデータ受け渡しの窓口となり、その番地に読み書きをすると、データの入出力ができる

# 電源装置

- ・ 現代のコンピュータシステムは、電気エネルギーを消費して、情報処理を行う自動機械である
  - 各構成要素の消費電力を十分まかなえる電源が必要
  - 電源が不安定になると、突然誤動作することもある
  - サーバマシンの場合は、無停電電源(UPS: Uninterruptible Power Supply)を装備して、停電時の信頼性を確保することが多い
  - 携帯機器では、電池の持続時間や充電時間が重要となる
  - 放熱のためのファンの配置など、空気の流れ・熱設計も重要
  - データセンタや通信施設では冷却のための電力が無視できない  
(例) 石狩市に巨大データセンタを設置 → 冷却コストが小さい
- ・ (参考) 人間は1人あたりおよそ100Wを消費している
  - 消費電力あたりの計算速度で勝負をする競技会も最近増えている



# ソフトウェアの構成要素

- 機械語
  - 計算機的基本的な動作の単位
- ファームウェア
  - ハードウェアの接続や機械語の命令セットを設定するための特別なプログラム。普段は表には出てこない
- オペレーティングシステム
  - 計算機的基本的な動作(入出力装置の制御、ファイル管理、プロセス管理など)を行うため常時動作する特別なプログラム
- コンパイラ／インタプリタ
  - 人間が読み書き可能なプログラミング言語を、実行可能な機械語に自動変換、または直接解釈して実行するためのプログラム
- アプリケーションプログラム
  - 計算機にユーザが望む仕事をさせるための応用プログラム
- ソフトウェアライブラリ
  - よく使用される基本的な操作を集めたプログラム群

# 今回のまとめ

## 計算機アーキテクチャとは

- 現代社会におけるコンピュータシステム
- 計算機の利用形態の分類とその特性
- アーキテクチャの評価尺度とトレードオフ
- ハードウェアの構成要素
  - レジスタ、演算器、プログラムカウンタ、命令デコーダ、メモリ(主記憶)、バス、ディスク(2次記憶)、入出力装置
- ソフトウェアの構成要素
  - 機械語、ファームウェア、オペレーティングシステム、コンパイラ／インタプリタ、ソフトウェアライブラリ、アプリケーションプログラム