コンピュータ基礎(6)

6 章 中央処理装置(pp. 196-241)
ただし，機械語命令(pp.205-214近辺)を除く

この章で学習すること

- 6.1 中央処理装置の仕組み
  - 構成要素：制御装置・演算装置
  - クロック，バス，命令について
- 6.2 制御装置
  - 命令実行の制御，順序
  - レジスタの種類と用途
- 6.3 算出論理演算装置
  - 扱うことが出来るデータの種類と，演算機構
- 6.4 CPU の入出力制御
  - パラレルバスとシリアルバス
  - 直接入出力と DMA
- 6.5 CPU関連アーキテクチャ
  - 高速化の仕組み（キャッシュ，パイプライン，並列処理）
  - マイクロプログラム，CISC, RISC

コンピュータの５大機能（再掲）

- 入力
  - 処理すべき情報の入力．操作.
  - キーボード，カメラ，バーコードリーダ，・・
- 記憶
  - 処理途中のデータを記憶する（主記憶装置）
  - 将来に備えてデータを保管する（補助記憶装置）
- 制御
  - 場合分け，条件判断などを行って処理を切り替える
- 演算
  - 加減乗除などの計算をする
- 出力
  - 計算結果を出力する．活用する.

コンピュータの構成要素（再掲）

- 中央処理装置（CPU, Central Processing Unit）
  （第６回の授業で詳しくやります）
  - 命令を主記憶装置から読み込んで解釈，実行する．
  - 四則演算や制御（条件判断）を行う．
- 記憶装置（第５回の授業で詳しくやります）
  - 主記憶装置：メインメモリ．計算機が動作している間に，処理途中のデータを一時的に記憶する．普通，電源を切ると内容が消えてしまう（揮発性）．
  - 補助記憶装置：ハードディスクなど．主記憶装置よりも大容量で，処理結果を長期的に記憶するために用いられる．電源を切手も内容は消えない（不揮発性）．
- 入出力装置（第４回の授業で詳しくやります）
  - パソコンであればマウスやキーボード，ディスプレイ．
  - 家電機器の制御や画面表示なども含む．

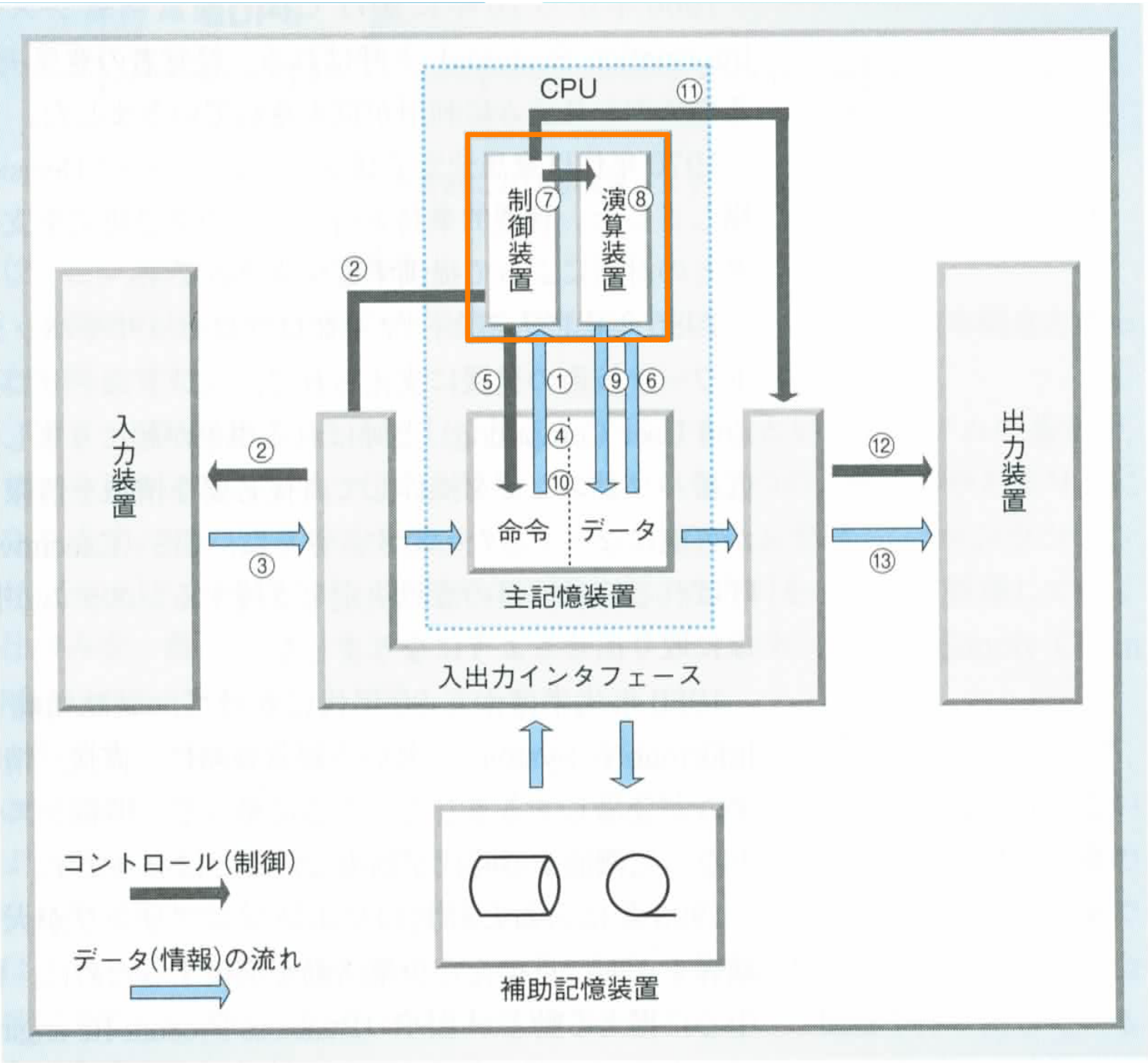
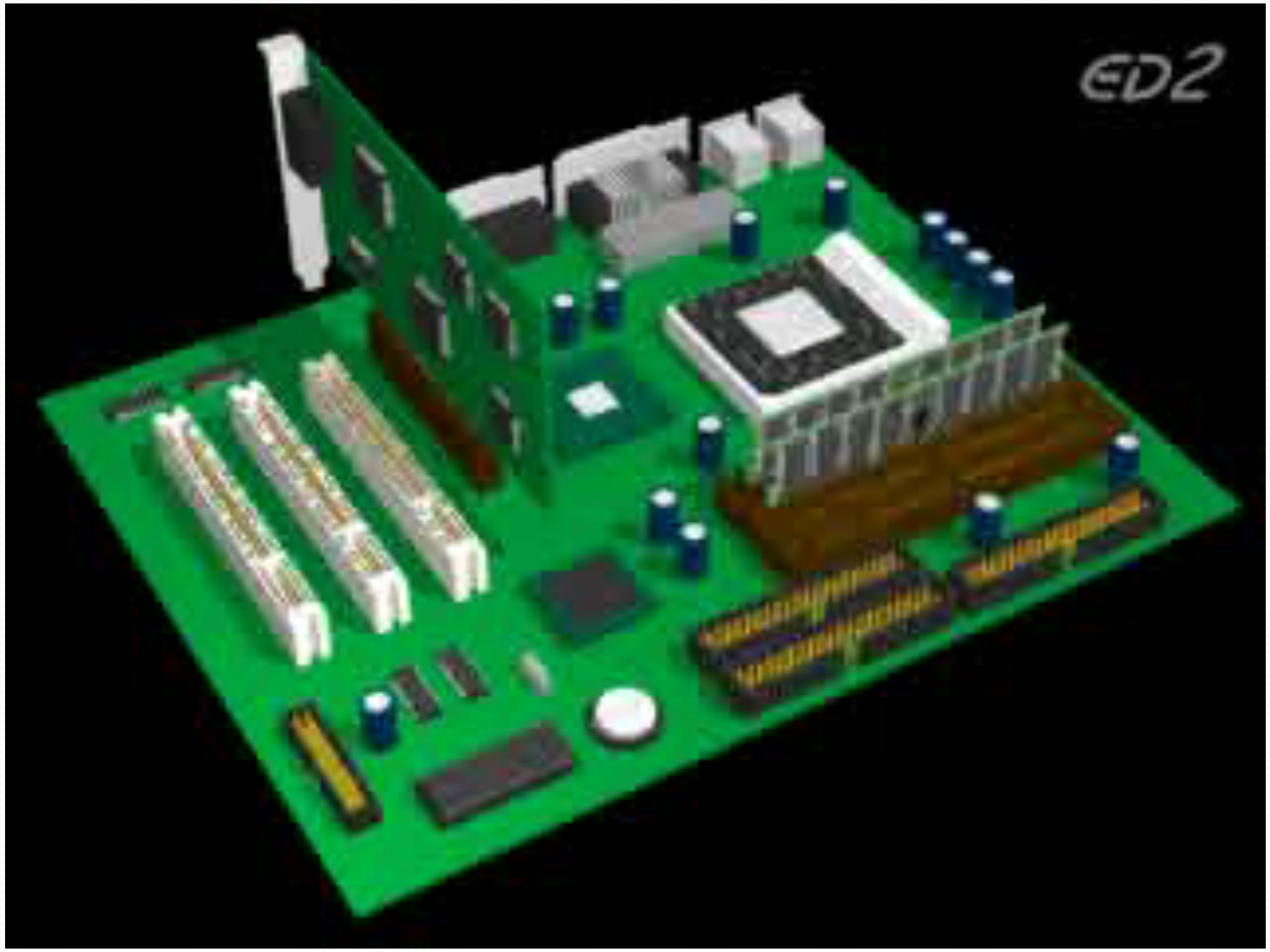
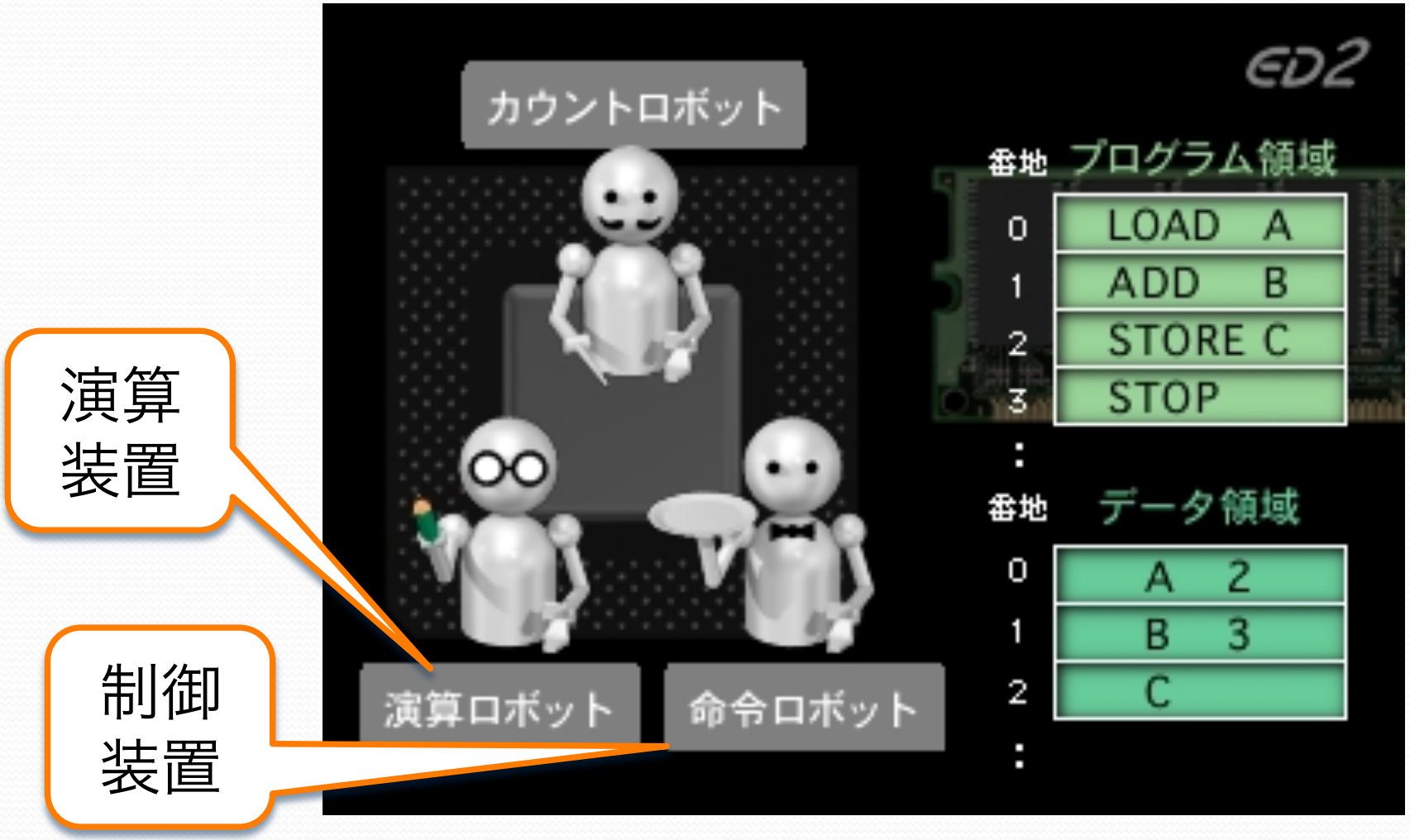


図 1-48　データの流れとコントロールの流れ

CPU の動作（ムービー）

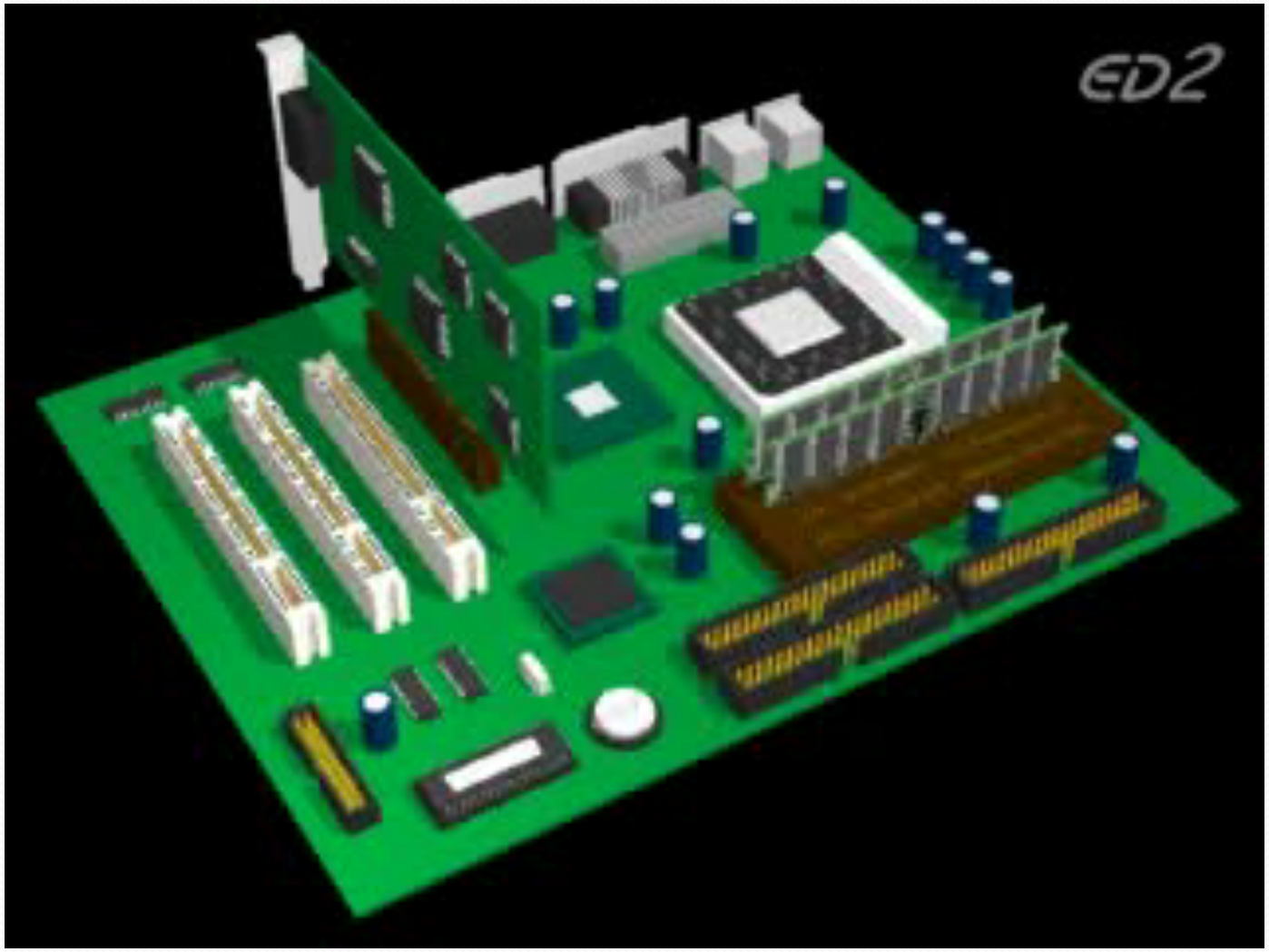


制御装置と演算装置



- 「計算を行う部分」だけでは，複雑な処理は出来ない．プログラム（命令）を解釈する部分が必要．

クロック



- コンピュータ内の動作のタイミングを取る仕組み

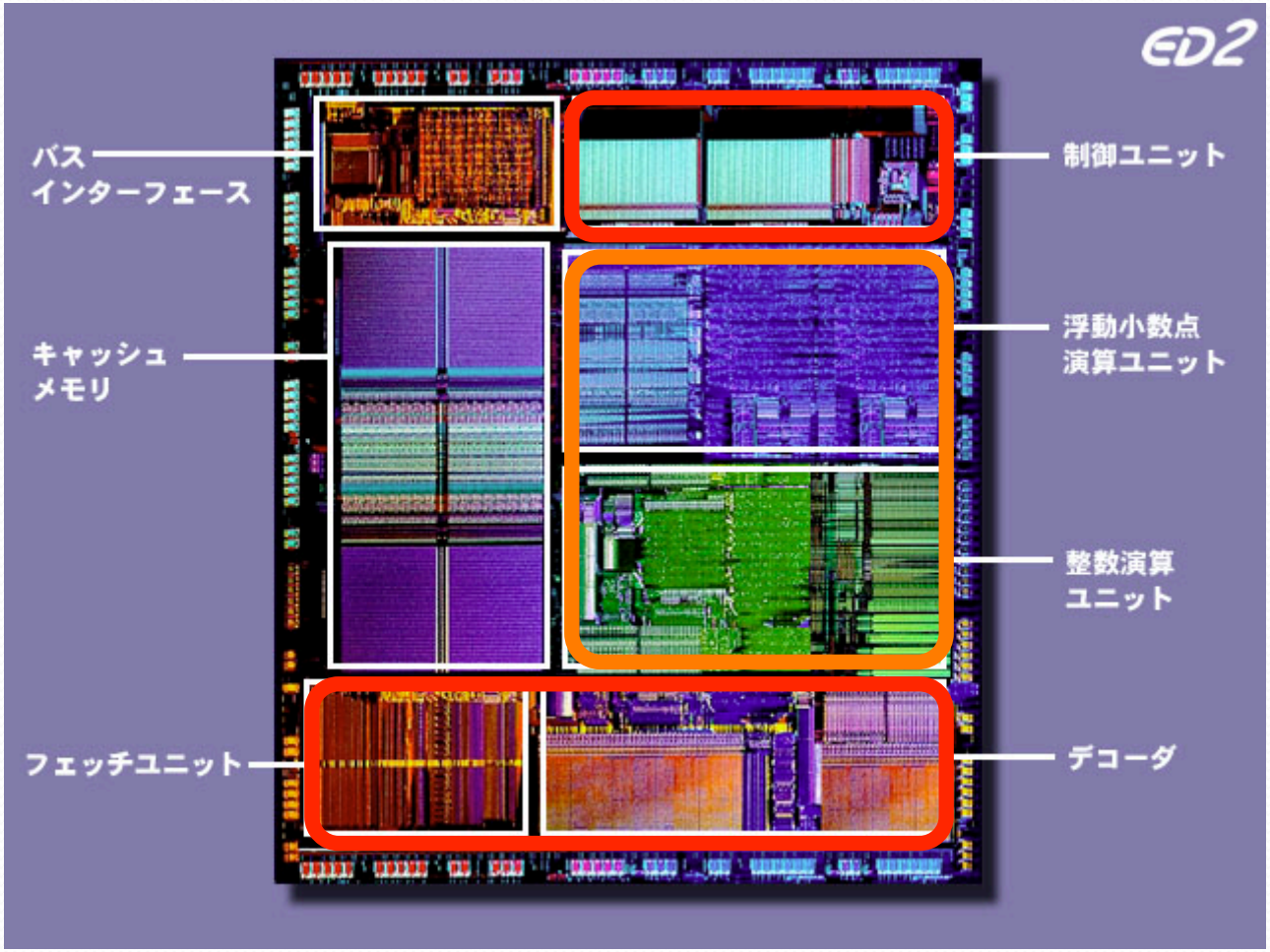
クロックとレジスタ

クロック

計算途中
結果の
保存場所
（レジスタ）

- クロックに合わせて命令の読み込み，解釈，実行が行われる．
- 計算途中結果はCPU内部のレジスタに蓄えられる．

CPU

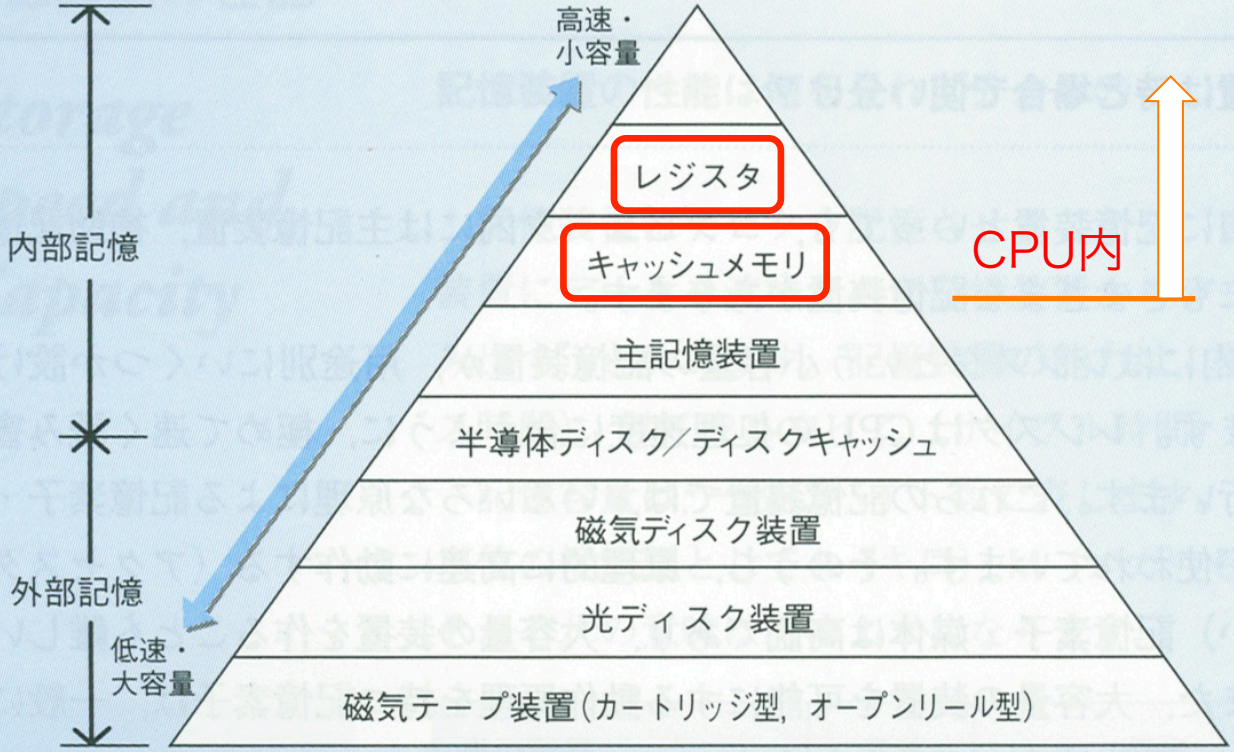


制御装置

演算装置

記憶階層

重要!!



アクセス時間	記憶容量
数百ps～数ns	～数バイト
数ns～数十ns	数十～数百Kバイト
数十～数百ns	数M～数十Gバイト
100μs～数ms	数百M～数百Gバイト
数十ms	数百M～数百Gバイト
100ms～数百ms	数百M～数百Gバイト
数s	数百M～数Gバイト／巻

注：カートリッジテープライブラリ装置では，アクセス時間が10秒程度，記憶容量は数十テラバイトです。

図 3-6　記憶階層の構造と特性

- 大容量のメモリほど遅い
  - 速度と記憶容量の両立のため，階層化されている

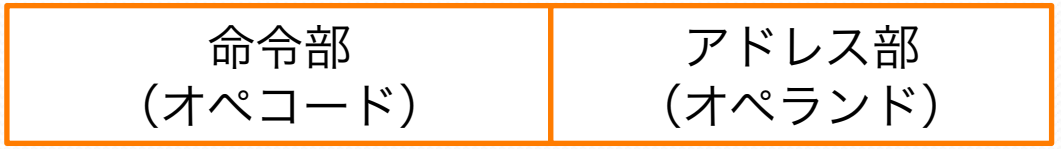
CPUの仕組み

- 制御装置
  - 主記憶装置から命令を取り出し・解釈して，他の装置（演算装置など）を動作させる
- 演算装置
  - 四則演算，論理演算，大小比較などを行う.
- レジスタ
  - 計算の途中結果や，処理対象のデータが主記憶装置のどこにあるか（データのアドレス）を記憶する.
  - CPU内部に設けられ，非常に高速だが小容量.
- クロック
  - 計算機の動作のタイミングを取る信号.
- バス
  - 計算機外部とデータのやりとりをする信号線.

CPUの動作と命令

- 命令とは？
  - 高級言語（例えばC言語）で作ったプログラムをコンパイルすると，実行ファイルができる．この中身は，CPUが直接解釈できる命令（機械語）である．
  - 機械語は，それぞれの単純な命令が二進数で表されている．
  - 変数名などの「名前」は全て，数値に置き換えられている．
    - int a, b, c; とすると，a, b, c という変数に対応するメモリの番地（アドレス：例えば 100, 104, 108 ）が決められる．
    - 例えば，C言語のプログラムで c = a + b; とすると，次のような機械語命令が作られる．
      - 100番地からデータを読み出してレジスタに入れる．
      - 104番地からデータを読み出してレジスタの値に加える．
      - レジスタの値を108番地に格納する．

命令の仕組み



- 命令部
  - 命令の種類を数値で表す．例えば，メモリからデータを読み出すのが 0x10, メモリへのデータの書き出しが0x20, 足し算が 0x30, 引き算が 0x40など.
- アドレス部
  - データを読んだり書いたりする対象のメモリのアドレスを表す.
  - 命令によっては，計算する値そのものだったり，レジスタの番号だったりもする.

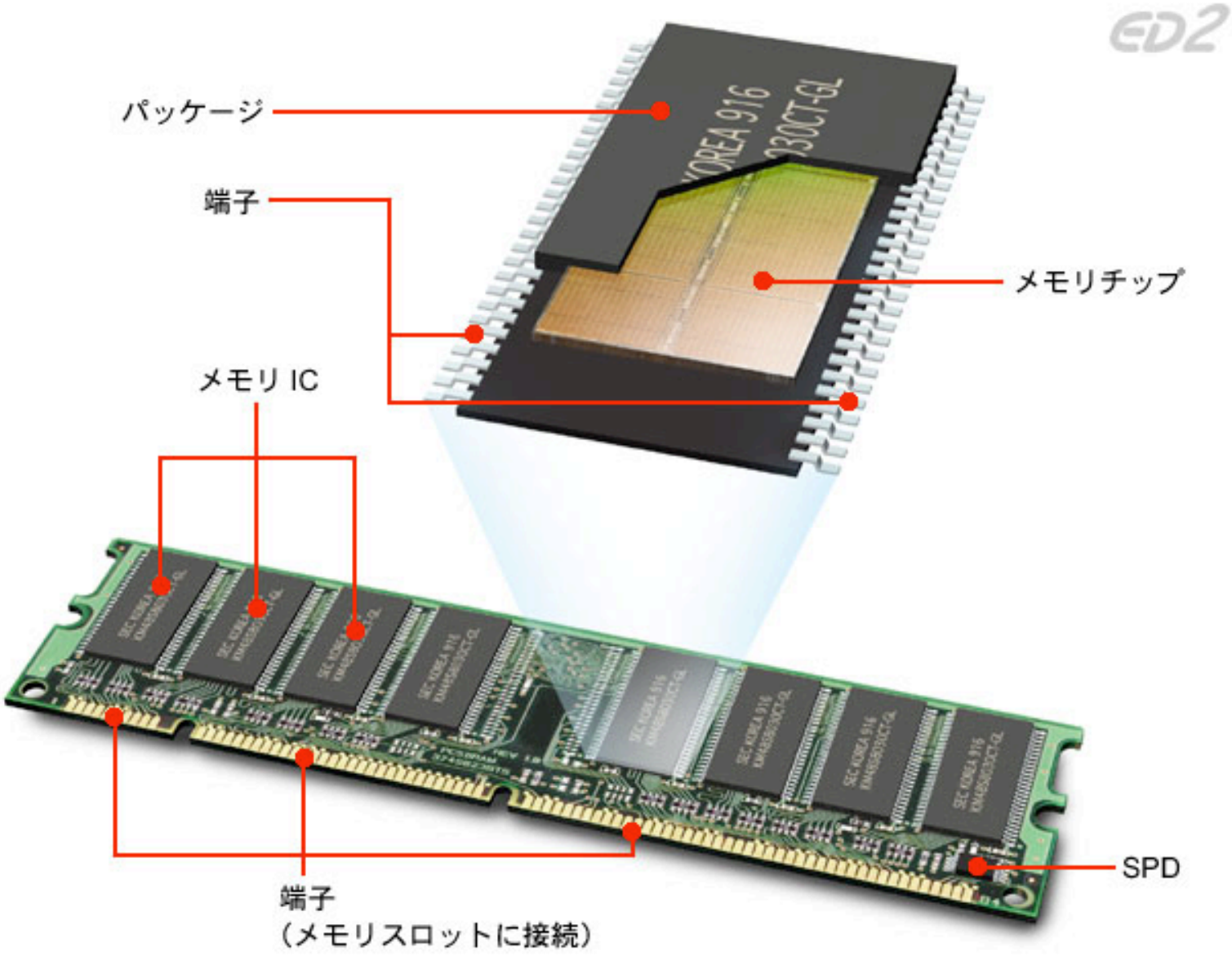
命令の読み出し

- 命令アドレスレジスタ（プログラムカウンタ）
  - 現在，実行中の命令がメモリのどこにあるかを常に記憶している，特別なレジスタ.
- CPUの動作
  1. [命令取り出し（フェッチ）]命令アドレスレジスタに記憶されているアドレスから，命令を読み出す.
  2. [命令解釈（デコード）]命令によって，CPUのどの回路を働かせるかを決める.
  3. [命令実行]メモリにデータを読み書きしたり，計算を行ったりする.

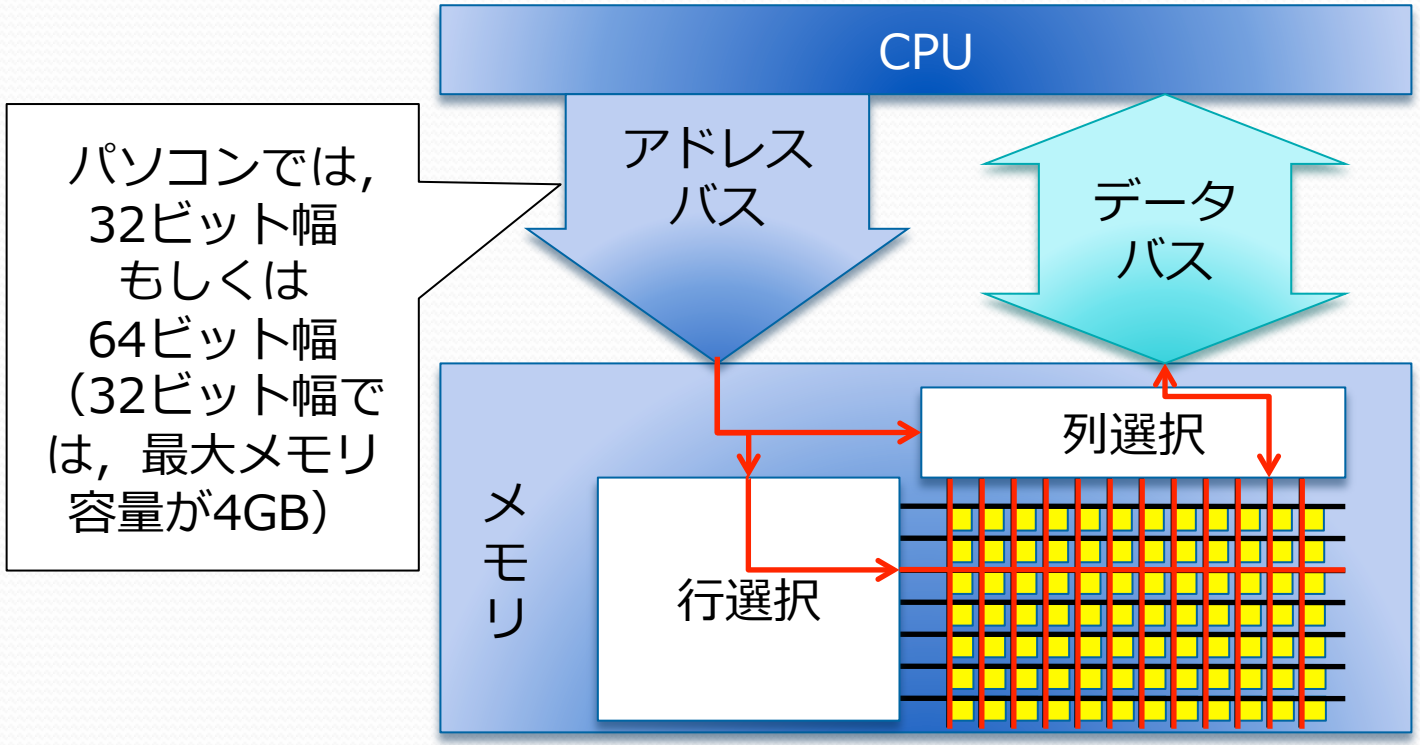
CPUの入出力制御

- 計算機のデータ信号線をバスと呼ぶ
  - CPUと直接つながっているもの：内部バス．CPUと，メモリや入出力装置をつなぐ．それ以外は外部バス．
  - パソコンに内蔵する拡張カードの端子をシステムバスと呼ぶ．共通規格化されている．
- パラレルバスとシリアルバス
  - パラレル（並列）バス：複数の信号線で同時にデータを送る．8bitのデータなら，8本の信号線で送るなど．
    - 例：PCI, SCSI, GP-IB, ATA など．
  - シリアル（直列）バス：１本の信号線に1bitずつ順次送る．最近は高速化が著しい．
    - 例：RS-232C, USB, IEEE1394, Serial-ATA など

メモリ(RAM)の構造



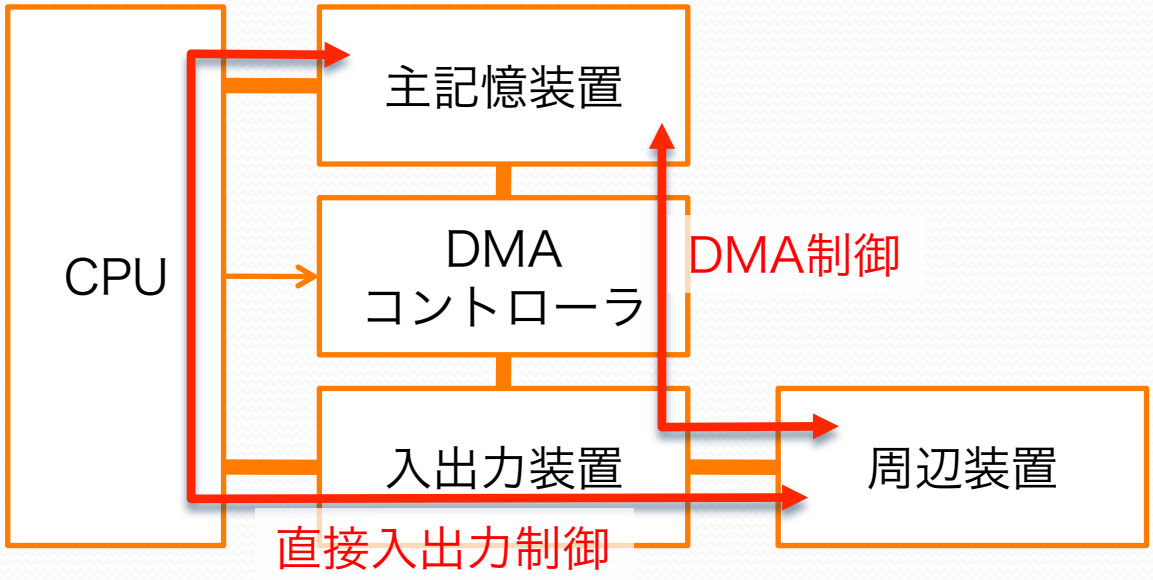
アドレスバスとデータバス



- アドレスは，CPUからメモリへの一方通行（アドレスバス経由）
  - アドレスバスの幅が，搭載できる最大メモリ容量を決める
- データは，双方向（データバス経由）
  - データバスの幅が，同時に読み書き出来るビット数を決める
  - 通常，ワード幅になっている
- メモリ内部では，一列が一斉に読み出し・書き込みされる

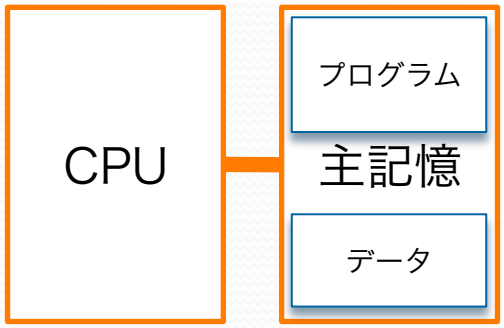
入出力制御方式

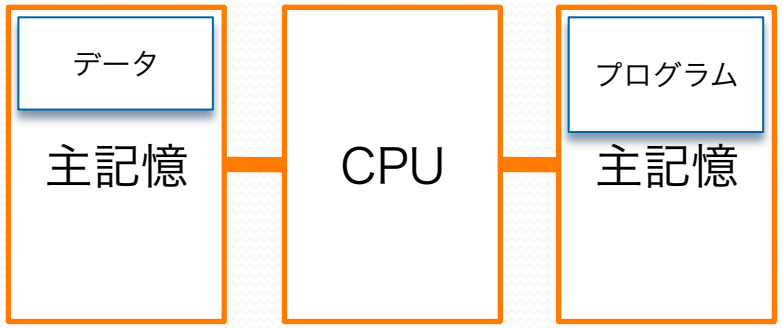
- 直接入出力制御方式
  - CPUが直接，命令によって一回一回の入出力を行う．
  - 入出力中は他の処理が出来ない．そのため遅い．
- DMA (Direct Memory Access)制御方式
  - DMAコントローラという装置によって，周辺機器からのデータを直接，主記憶装置に出し入れする．



CPUアーキテクチャ

- コンピュータアーキテクチャとは？
  - Architecture：建築のこと.
  - コンピュータアーキテクチャは，なるだけ高速・大容量で使いやすい計算機を安価に実現するための設計思想や，構成様式のこと.
- 例



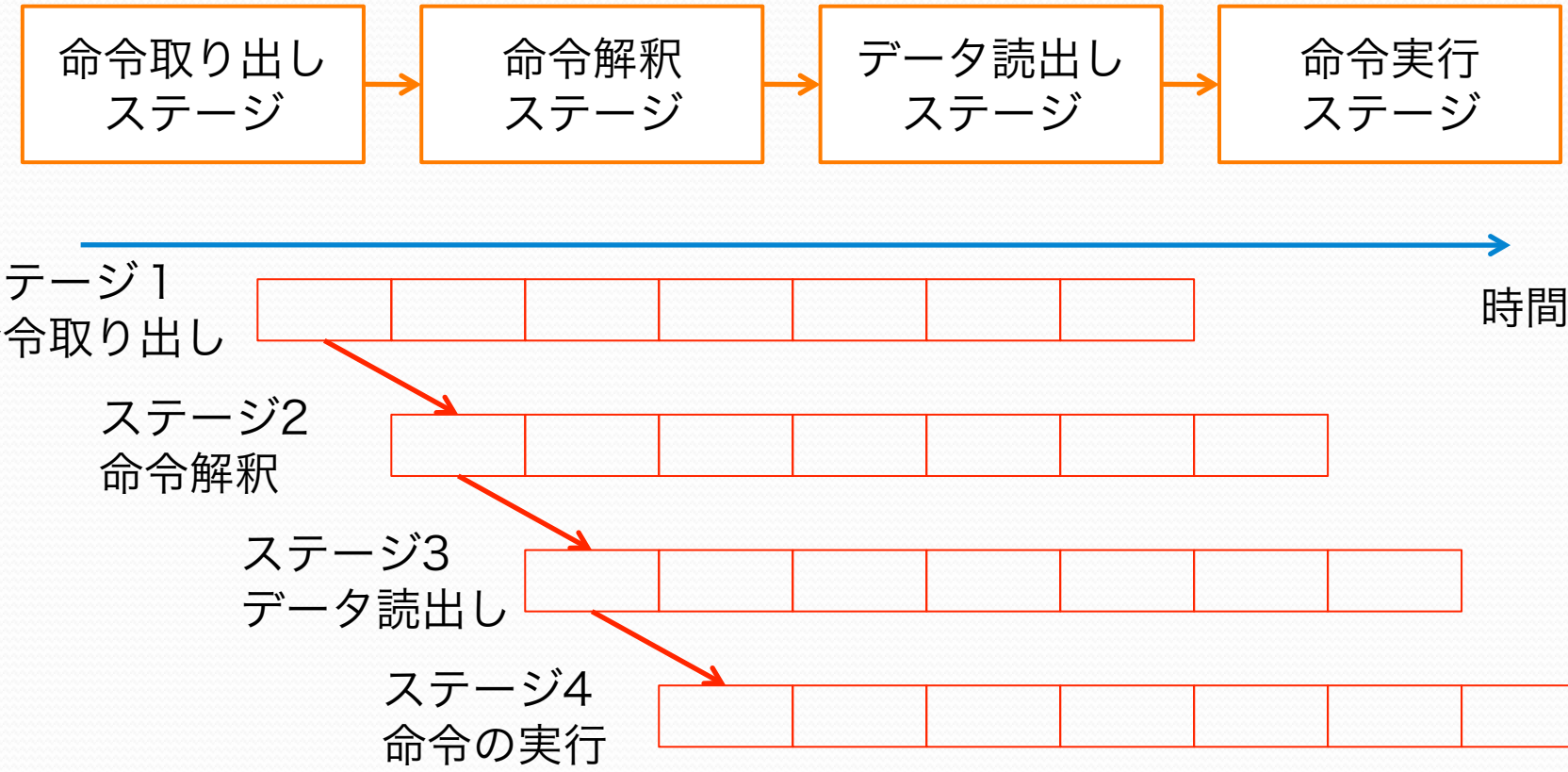


- ノイマン型コンピュータ：プログラムとデータが同じメモリに同居している
  - 利点：補助記憶装置からプログラムを読み込んで入れ替えることなどが簡単.
  - 欠点：メモリ上のプログラムとデータに交互にアクセスするため，処理速度が低下しやすい.
- ハーバードアーキテクチャ：プログラム用とデータ用のメモリが分かれている
  - 利点：プログラム側のメモリのアクセスは順序性が高い．また，同時アクセス可.
  - 欠点：コンピュータ自身によるプログラムの入れ替えに難がある.

高速化の工夫

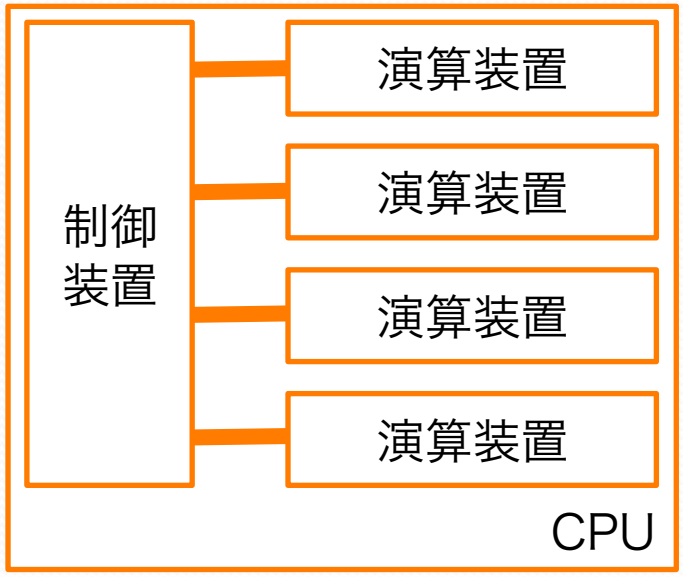
- キャッシュ（前回講義）
  - CPU内に置いた高速なメモリにデータを蓄えておき，メモリアクセスの遅延の影響を避ける．
- パイプライン制御（パイプライン処理）
  - 処理をいくつかの段階（ステージ）に分け，第１すてじの処理が終わり第２ステージに引き継ぐと，すぐに第１ステージは次の処理にとりかかる方式．
- 並列処理
  - 処理装置を複数用意し，同時に複数の処理を行う．
  - SIMD (Single Instruction Multiple Data)
    - １つの命令で複数のデータを同時に処理する
  - MIMD (Multiple Instruction Multiple Data)
    - それぞれ異なる命令で複数のデータを同時に処理する

パイプライン制御

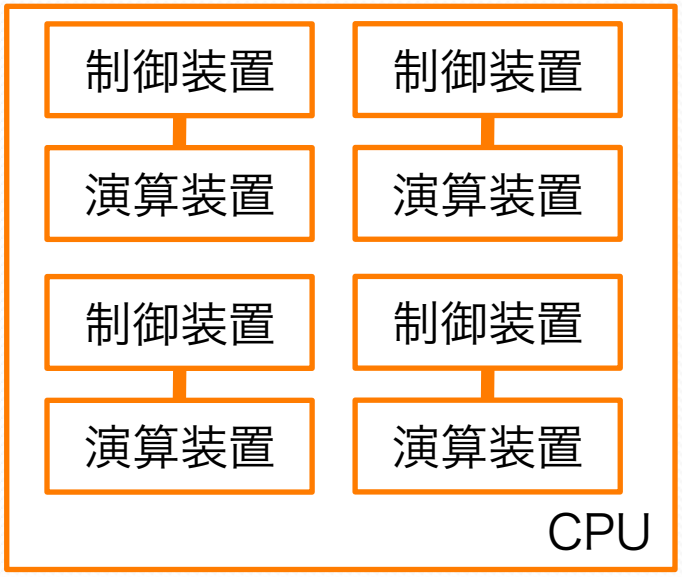


- ステージが多いほど速くなる
- 前回処理の結果を次の処理が使う場合（例えば前回処理の結果で次の処理が切り替えられる場合など），遅くなる

並列処理

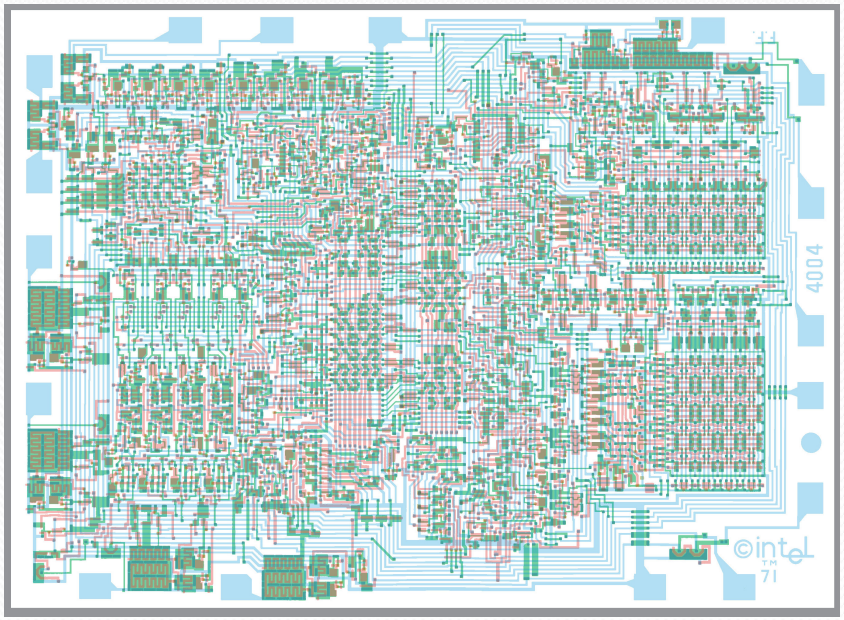


- SIMD型並列計算機：全ての演算装置が同じ命令を実行する.
  - 制御装置が１つでよい.
  - 行列計算（ベクトル同士の積）やマルチメディア処理（音声や画像の処理）に向いている.
  - グラフィックス描画装置(GPU)もSIMD型計算機である.
  - ベクトル型計算機とも呼ばれる.

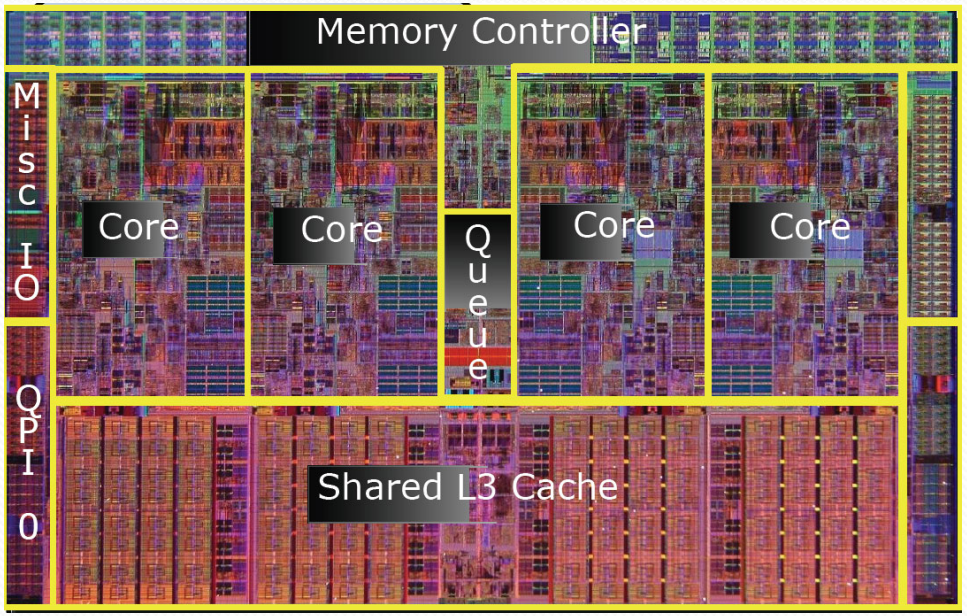


- MIMD型並列計算機：演算装置ごとに別の命令を実行できる.
  - 異なるプログラムを同時に実行するときなどに向いている.
  - 科学技術用の超大型の計算機を作る場合にもよく用いられる.
  - クラスタ型並列計算機とも呼ばれる.

CPUの高集積化



Intel 4004，1971年
2300トランジスタ
動作クロック：731kHz



Intel Core i7，2008年
7億3100万トランジスタ
動作クロック：3.33GHz

コア数のプログラムを同時実行できる

命令アーキテクチャ

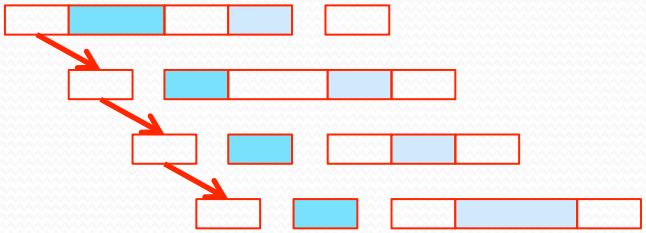
- CISC アーキテクチャ
  - Complex Instruction Set Computer
    複合命令セットコンピュータ
  - １つの命令で複雑な処理ができる
    - 例：加算などの計算により求めたアドレスからデータを読み出して，さらに四則演算などを実行することができる.
    - プログラムを少ない命令数で実現することができる.
- RISC アーキテクチャ
  - Reduced Instruction Set Computer
    縮小命令セットコンピュータ
  - １つ１つの命令が非常に単純
    - 例：アドレス計算，メモリからのデータの読み出し，演算処理，データの書き出しなどがそれぞれ別の命令
    - １つの命令の実行速度を速くできる.

CISCとRISC

命令部	アドレス部
命令部	アドレス部
命令部	アドレス部
命令部	アドレス部
命令部	アドレス部
・	・
・	・

命令部	アドレス部
命令部	アドレス部
命令部	アドレス部
命令部	アドレス部
命令部	アドレス部
・	・
・	・

- CISC
  - 命令やアドレス部の大きさがまちまち
  - １つの命令実行にかかる時間が一定でないため，パイプライン効率が落ちる
  - 回路が大規模になってしまい，並列化が難しくなる
  - 開発に時間がかかる



- RISC
  - 命令やアドレスの長さが一定
  - パイプライン処理に向いている
  - プログラムの容量が大きくなりがち

- 最近は，命令セットはCISCだが，CPU内部で命令を自動的に細分化してRISC計算機で実行するものが主流になっている.