

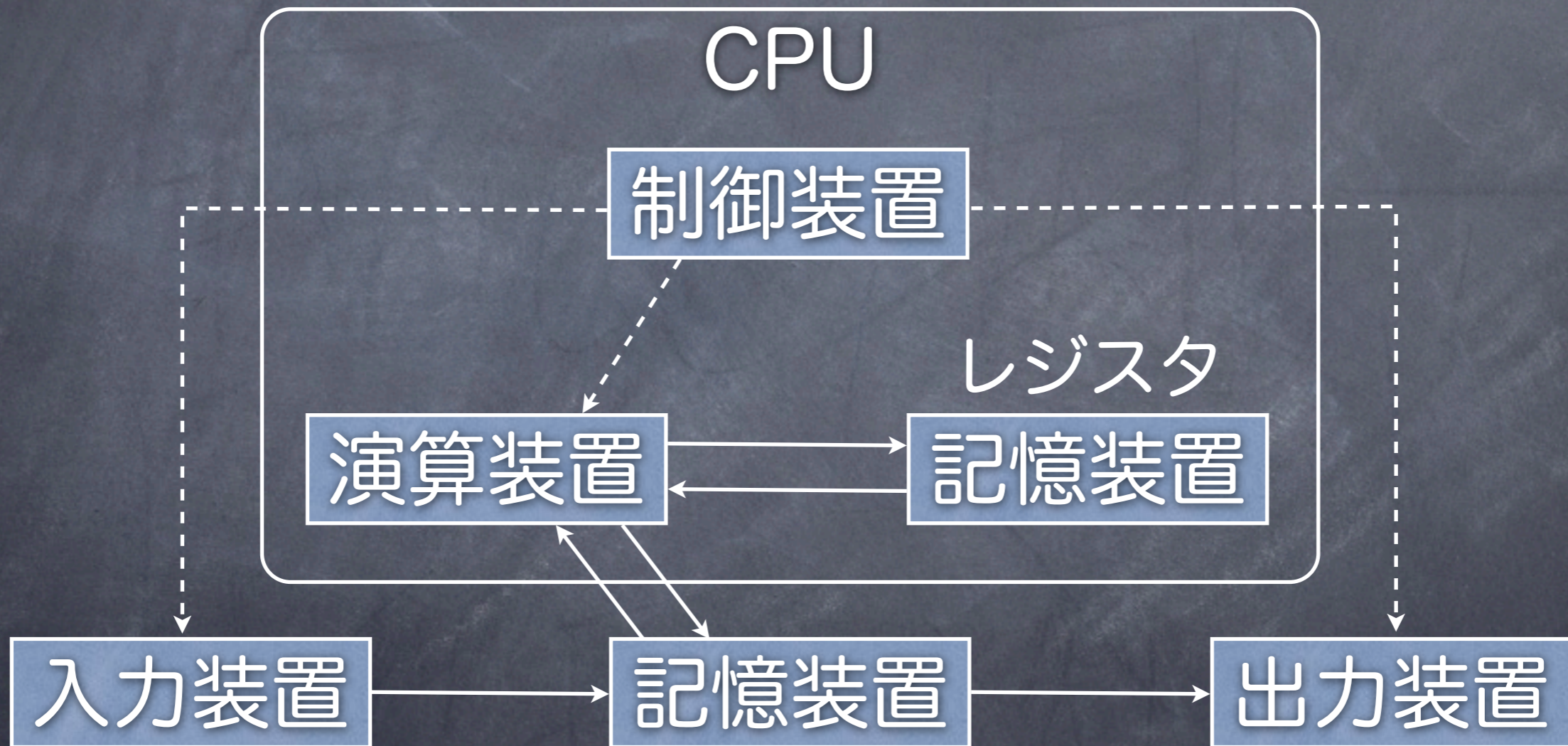
情報科学

(1) コンピュータにおける情報表現

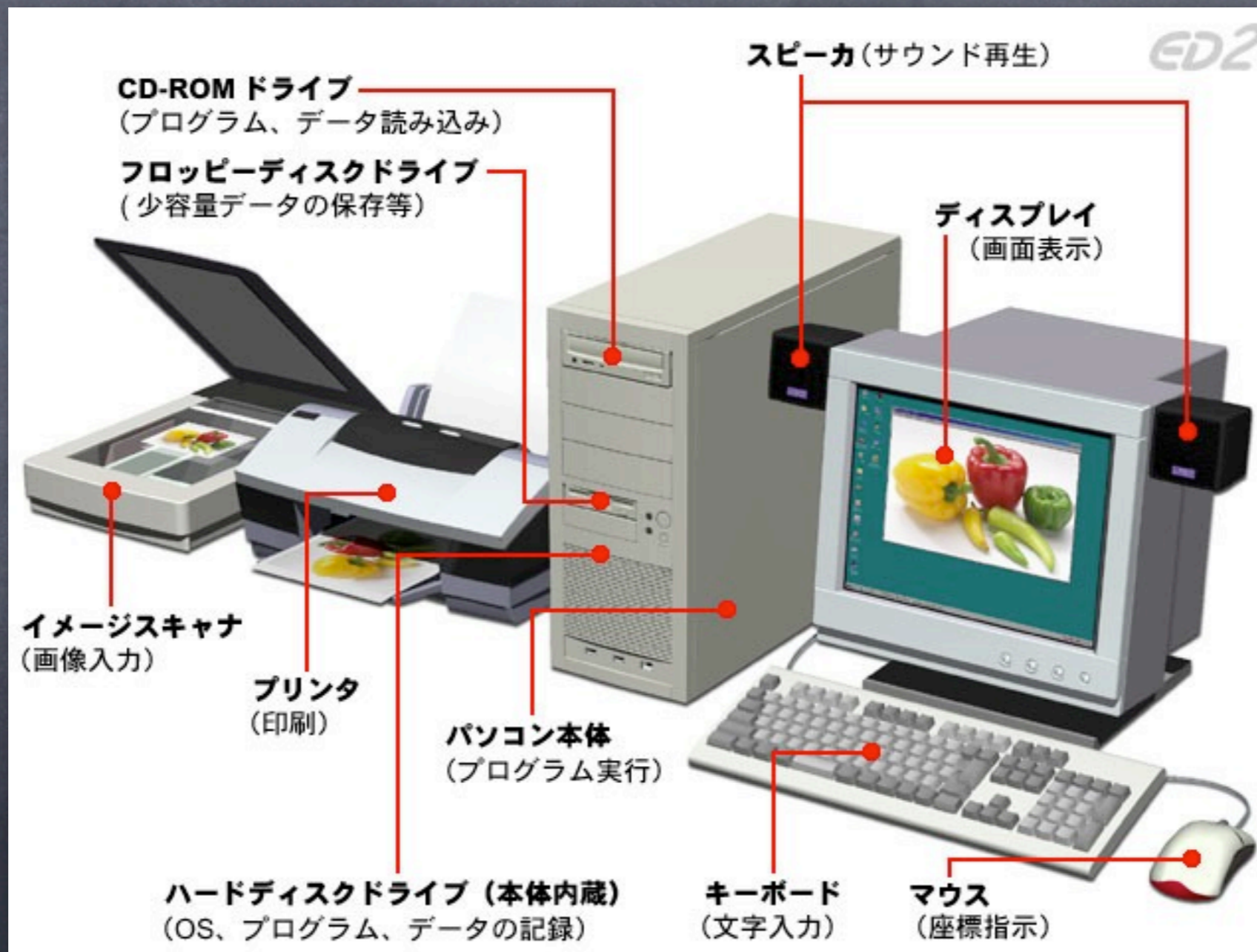
コンピュータの5大装置

- 制御装置
 - 演算装置
- CPU
- 記憶装置 → RAM (メモリ), HDD
 - 入力装置 → キーボード, マウスなど
 - 出力装置 → ディスプレイ, プリンタ

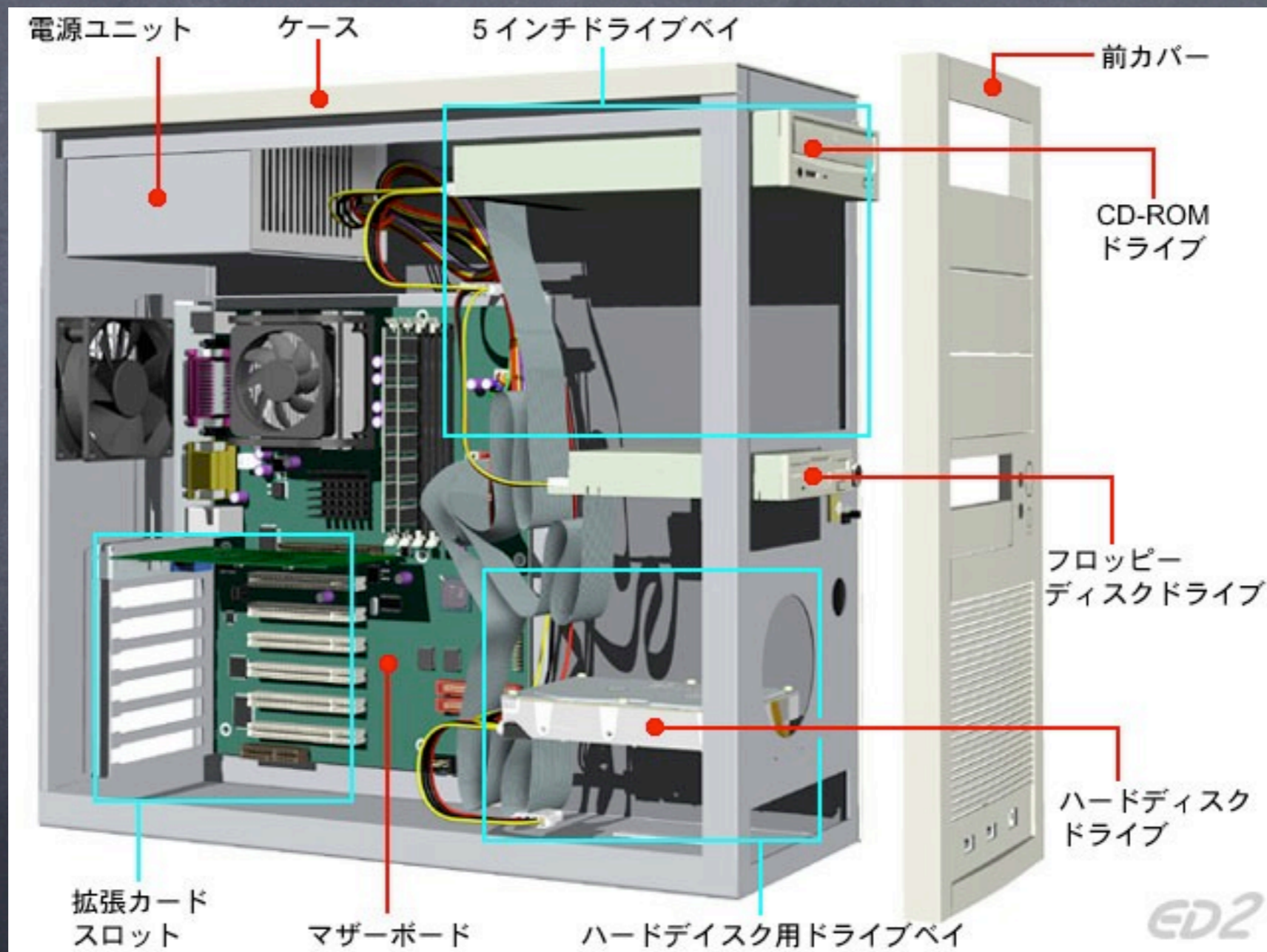
コンピュータの構造



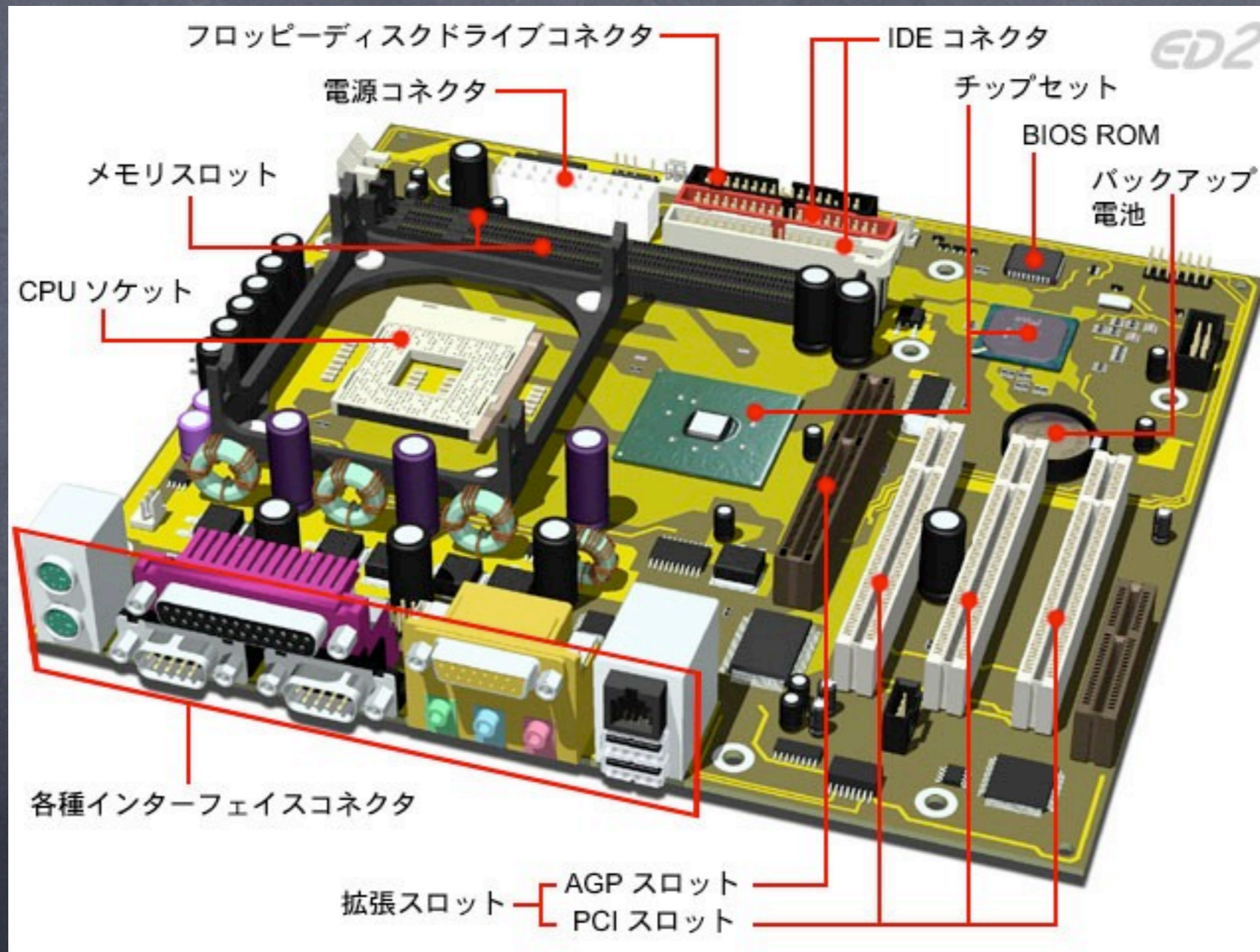
コンピュータの構成



コンピュータの中



マザーボード



コンピュータの動作



記憶装置

- 主記憶装置 = CPUが直接読み書きできる
 - RAM(Random Access Memory)
 - ROM(Read Only Memory)
- 補助記憶装置
 - 磁気ディスク装置 (HDD)
 - 光ディスク装置 (CD, DVD, BD)

主記憶装置の構造

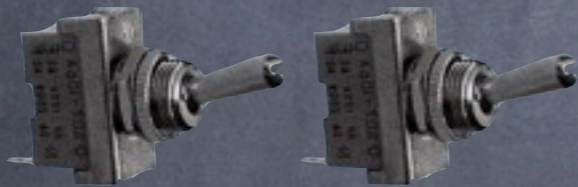


主記憶装置の構造

「スイッチ」が並んだようなもの



1つ：ONとOFFの2状態



2つ：4状態



3つの時は？



4つの時は？

1GBのメモリ = 8,589,934,592個のスイッチ

情報量の単位

- 1ビット(bit) = 1個のスイッチで表現できる量 = 2状態
- nビット = n個のスイッチで表現できる量 = 2^n 状態
- 8ビット = 1バイト(byte) (1Bとも表現)
- $2^{10}B \doteq 1000B = 1KB$
- $2^{20}B \doteq 1000KB = 1MB$
- $2^{30}B \doteq 1000MB = 1GB$
- $2^{40}B \doteq 1000GB = 1TB$

m状態を表現するのに
必要なビット数は？

コンピュータ内部におけるデータ表現

- コンピュータの扱うデータ
 - 数値, 文字, 画像, 音声 . . .
- 「スイッチ」の組み合わせでデータを表現
 - 対応表 (2進数, 文字コード表など)

数値の10進数表現

- 10進数：各桁は0~9の10状態を表現

5643.012

5×重み 10^3

3×重み 10^0

1×重み 10^{-2}

n桁目の重みは 10^n

数値の2進数表現

- 2進数：各桁は0~1の2状態を表現



n桁目の重みは 2^n

10進数から2進数への変換

- 整数部：2で割って余りを並べる
- 小数部：2を掛けて整数部を並べる

$$10進数 6.375 = 2進数 110.011$$

整数部 = 110

$$6 \div 2 = 3 \dots 0$$

$$3 \div 2 = 1 \dots 1$$

$$1 \div 2 = 0 \dots 1$$

小数部 = 011

$$0.375 \times 2 = 0.75$$

$$0.75 \times 2 = 1.5$$

$$0.5 \times 2 = 1.0$$

10進数からn進数への変換

- 整数部：nで割って余りを並べる
- 小数部：nを掛けて整数部を並べる

$$10\text{進数 } 11.375 = 8\text{進数 } 13.3$$

整数部 = 13	小数部 = 3
$11 \div 8 = 1 \dots 3$	$0.375 \times 8 = 3.0$
$1 \div 8 = 0 \dots 1$	

16進数

- 各桁は0～9, A～Fで16状態を表現
- 第 n 桁目の重みは 16^n
- 2進数との対応が分かりやすい
 - 2進数 0101 = 16進数 5
 - 2進数 1100 = 16進数 C
 - 2進数 0101 1100 = 16進数 5C

2進数の補数表現

- 2進数には「2の補数」と「1の補数」がある
- 1の補数：各桁を反転させる
 - 0101の「1の補数」は 1010
- 2の補数：1の補数に1を加える
 - 0101の「2の補数」は 1011

※一般にn進数には「nの補数」と「n-1の補数」がある

補数表現と演算

- 多くのコンピュータは負数=2の補数

例) $5 + (-5)$

$5 = 0101, -5 = 1011$

$0101 + 1011 = 10000$

下4桁だけ見れば $5 + (-5) = 0$ が成立

実数の固定小数点表現

- 負の数は「2の補数」表現
= 最上位を符号桁とする (0=正, 1=負)
- 小数点の位置は固定

コンピュータにおける数値表現

- 8桁単位の固定長が多い
 - 8桁（表現量8ビット）
 - 符号有り：-128~127
 - 符号無し：0~255
 - 16桁（同16ビット）
 - 符号有り：-32768~32767
 - 符号無し：0~65535
 - 32桁，64桁，128桁など

浮動小数点表現

- 桁数固定長の数値で大きな範囲の数を表現
 - 実数 $Y = M \times B^E$
 - M : 仮数部, B : 基数部, E : 指数部
- 同じ数でも複数の表現が可能
 - 例) 123000の浮動小数点表現
 - 123×10^3
 - 12.3×10^4
 - 1.23×10^5
 - 0.123×10^6

正規化：仮数部の表現を固定することで全体の表現を固定する

誤差

- 桁あふれ誤差：データの桁数超過による誤差
 - オーバーフロー：最大値を超える
 - アンダーフロー：最小値を下回る
- 丸め誤差：数値を有効桁数に納めるために発生する誤差
 - 切り捨て, 切り上げ, 四捨五入によって発生
- 桁落ち誤差：演算途中の有効桁数の不足による誤差
 - 絶対値のほぼ等しい数の演算で発生
- 情報落ち：大きい値と小さい値の演算で小さい値が消えること
- 打ち切り誤差：浮動小数点の演算を打ち切ることで発生
 - 例) 円周率を3.14とすることで発生する誤差

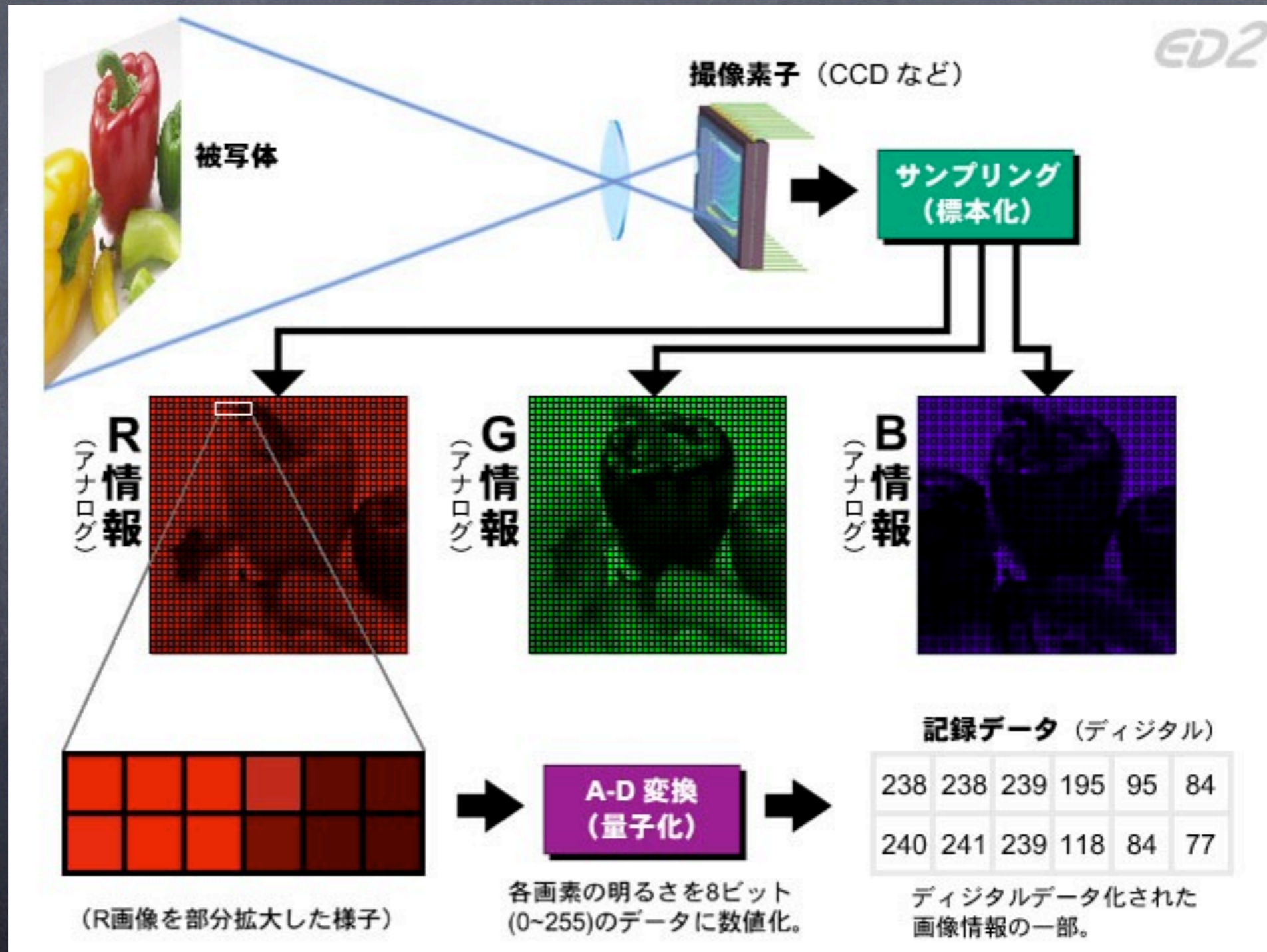
シフト演算

- シフト演算：桁をずらす演算（ n 桁ずらすと 2^n 倍）
例) 0001 1010 (26) を...
 - 右に1桁シフト 0000 1101 ($13=26 \times 2^{-1}$)
 - 左に2桁シフト 0110 1000 ($104=26 \times 2^2$)
- 論理シフト
 - そのままシフトして、あいた桁には0を入れる
- 算術シフト
 - 符号桁はそのまま
 - あいた桁には...
左シフト時は0, 右シフト時は符号桁と同じ数を入れる

文字コード表

- メモリの「スイッチ」の状態と文字の対応表
 - 英語（アメリカ）
 - ASCIIコード表：1文字を8桁で表現
 - 日本語：日本語1文字を16桁で表現
 - JIS, Shift-JIS, EUC-JP
 - Unicode：世界中の文字を単一のコード表で表現
 - UTF-16, UTF-8など

画像データ



音声データ

