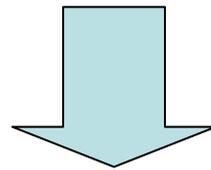


コンピュータ(計算機)

CPUとメモリ、
記憶装置と入出力装置

3章の概要

- 情報の表し方、情報の演算の仕方は1,2章で説明した
- では、情報の演算の指示のやり方は？
 - どのような形で演算を指定するのか？
 - どのような形で演算するデータを指定するのか？



- コンピュータの構成とそれへの演算の指示方法
- コンピュータの性能向上、性能評価、など

節概要

- コンピュータの構成
- 半導体と集積回路
- CPU(Central Processing Unit)の構成
 - 命令セット
 - 基本的な構成
 - アドレッシングモード
 - 高速化技術
- メモリアーキテクチャ
- 入出力装置
 - 補助記憶装置(磁気ディスク、光ディスク)
 - 入出力インタフェース
 - 代表的な入出力装置

コンピュータの種類と特徴

- スーパーコンピュータ
 - 大規模な科学計算技術計算用に設計
 - 最も高速・高性能なコンピュータ
- 汎用コンピュータ(メインフレーム)
 - 事務処理から技術計算までの多目的に利用できるよう設計された大型コンピュータ
 - 高い耐故障性や故障時の代替処理の迅速化を考えて設計
- ワークステーション
 - 高度な処理能力が求められる技術分野やネットワークサーバ分野で利用



Cray-1



ラックマウント型ワークステーション

コンピュータの種類と特徴

- パーソナルコンピュータ(パソコン)
 - 家庭やオフィスなどで多目的に利用されるコンピュータ
- 携帯情報端末(PDA: Personal Digital Assistant)
 - ノートパソコンよりも小型で携帯して持ち運ぶことが可能なコンピュータ
 - スマートフォンによって復権
- マイクロコンピュータ(マイコン)
 - 1つのチップに納められたコンピュータ
 - 家電製品などに組み込んで利用される
 - マイコンはマイクロコントローラの略にも



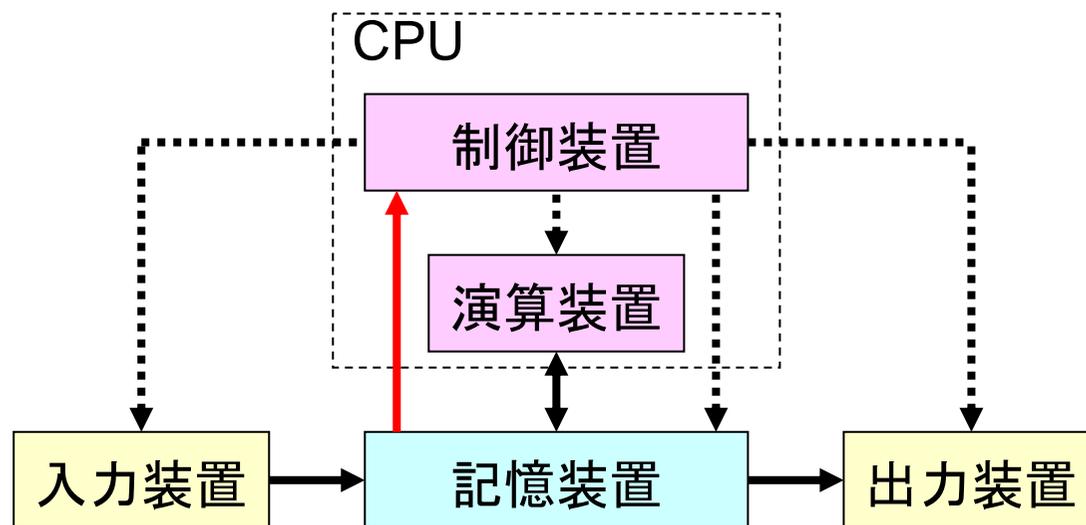
PDA(スマートフォン)



マイコン

3.1 コンピュータの構成

5大装置(機能): 入力、出力、記憶、制御、演算



- 実線はデータの流れ
- 破線は制御の流れ
- 赤線は命令の流れ

(1) コンピュータの構成

- 入力装置
 - データやプログラムを呼び込む
 - キーボード、マウス、タブレット
- 記憶装置
 - データやプログラムを記憶する。
 - 記憶装置には、主記憶装置と補助記憶装置がある。
 - 主記憶装置：コンピュータ内部にありCPUで実行するプログラムやデータを格納する。揮発性
 - DDR(2|3)-SDRAM
 - 補助記憶装置：実行前／後のプログラムやデータを格納。大容量低価格、不揮発性。
 - CD-ROM、DVD-ROM、BD-ROM、SSD、HDD、磁気テープ

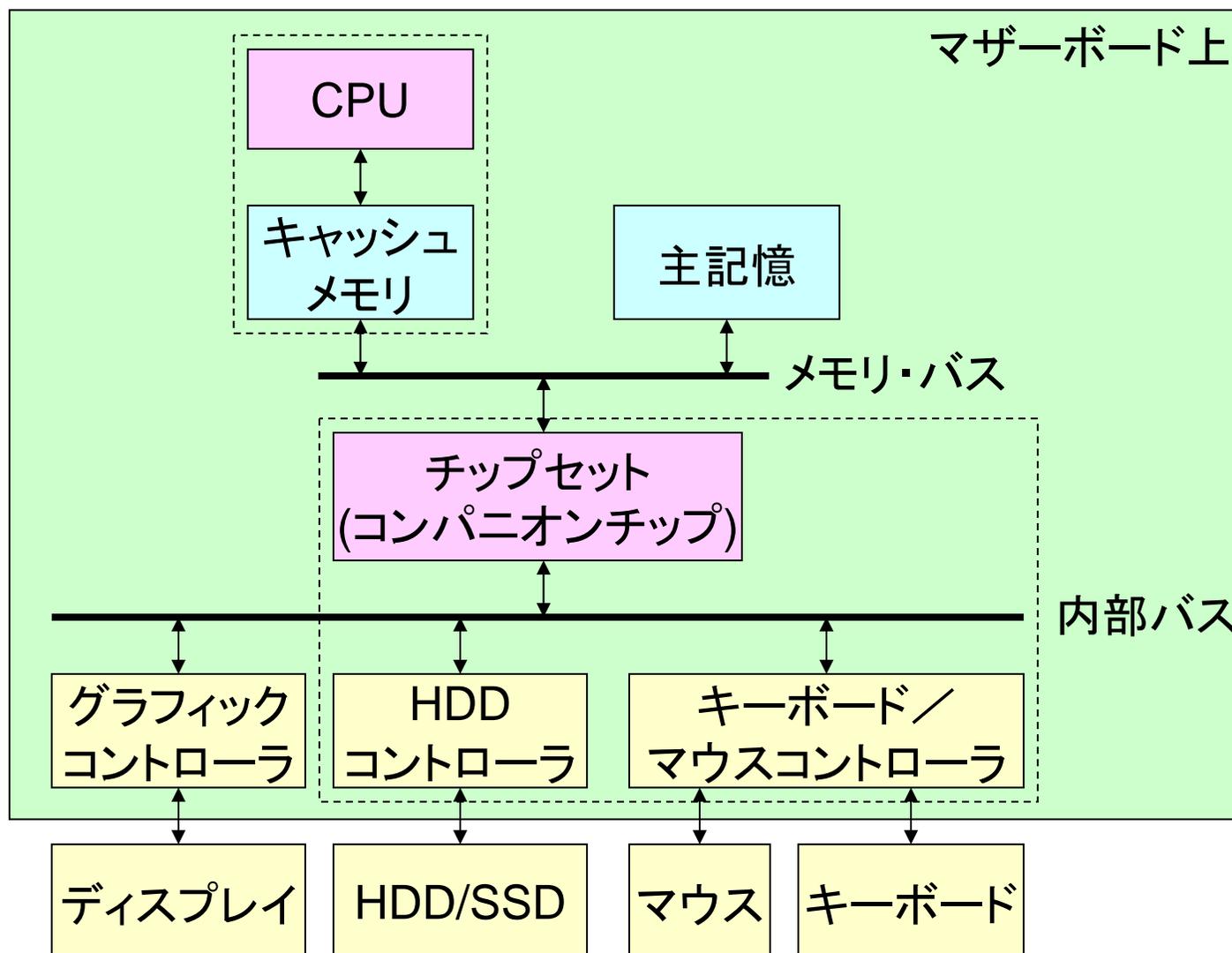
(1) コンピュータの構成

- 出力装置
 - 処理結果のデータを外部に取り出す
 - 例: ディスプレイ、プリンタ、スピーカー
- 演算装置
 - 計算、比較、判断などを行う
- 制御装置
 - 入力装置、記憶装置、出力装置、演算装置の制御を行う

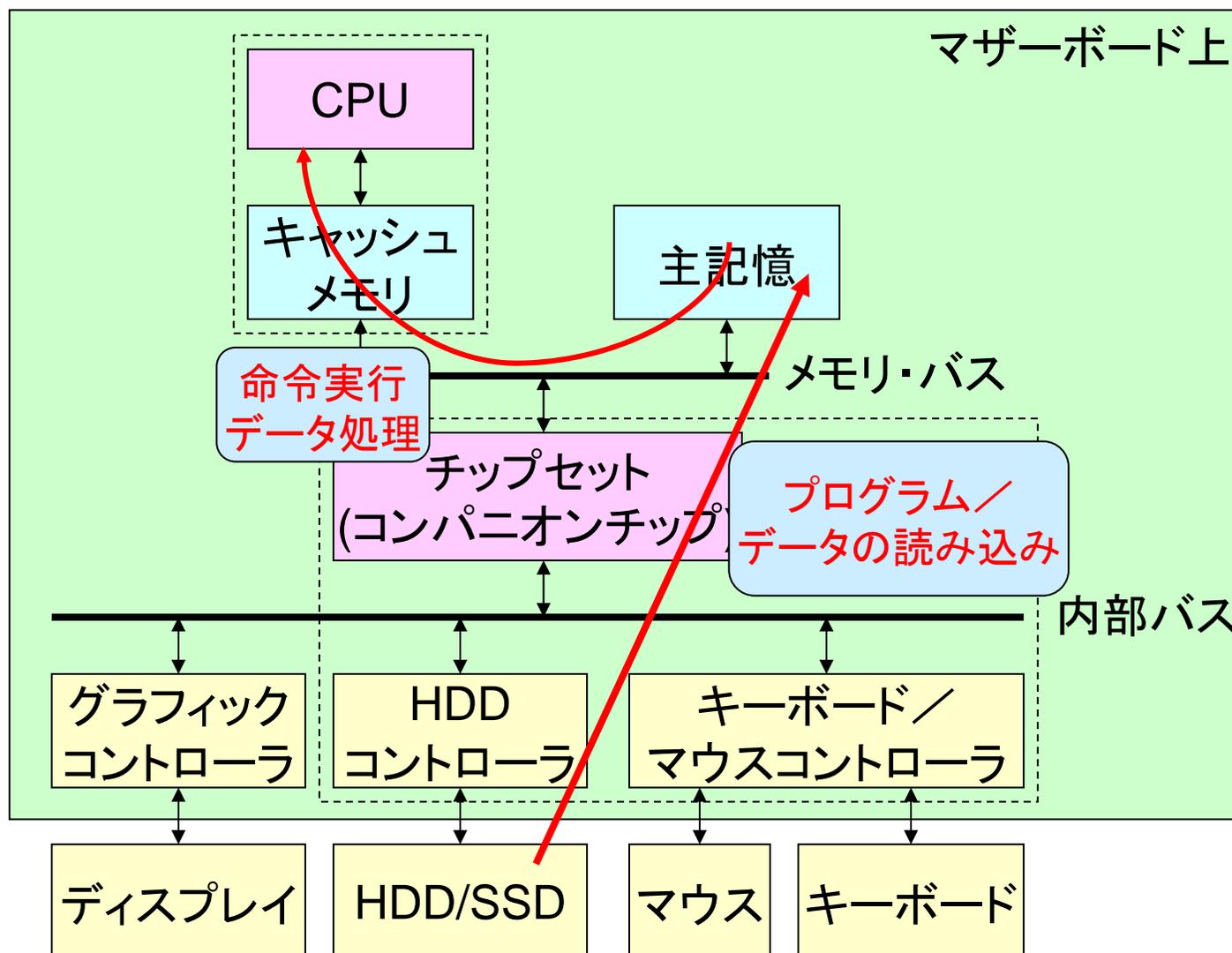


CPU(Central Processing Unit):
演算装置と
制御装置からなる

(2) コンピュータの基本的な構成



(2) コンピュータの基本的な構成



3.2 半導体素子(トランジスタ)とメモリ

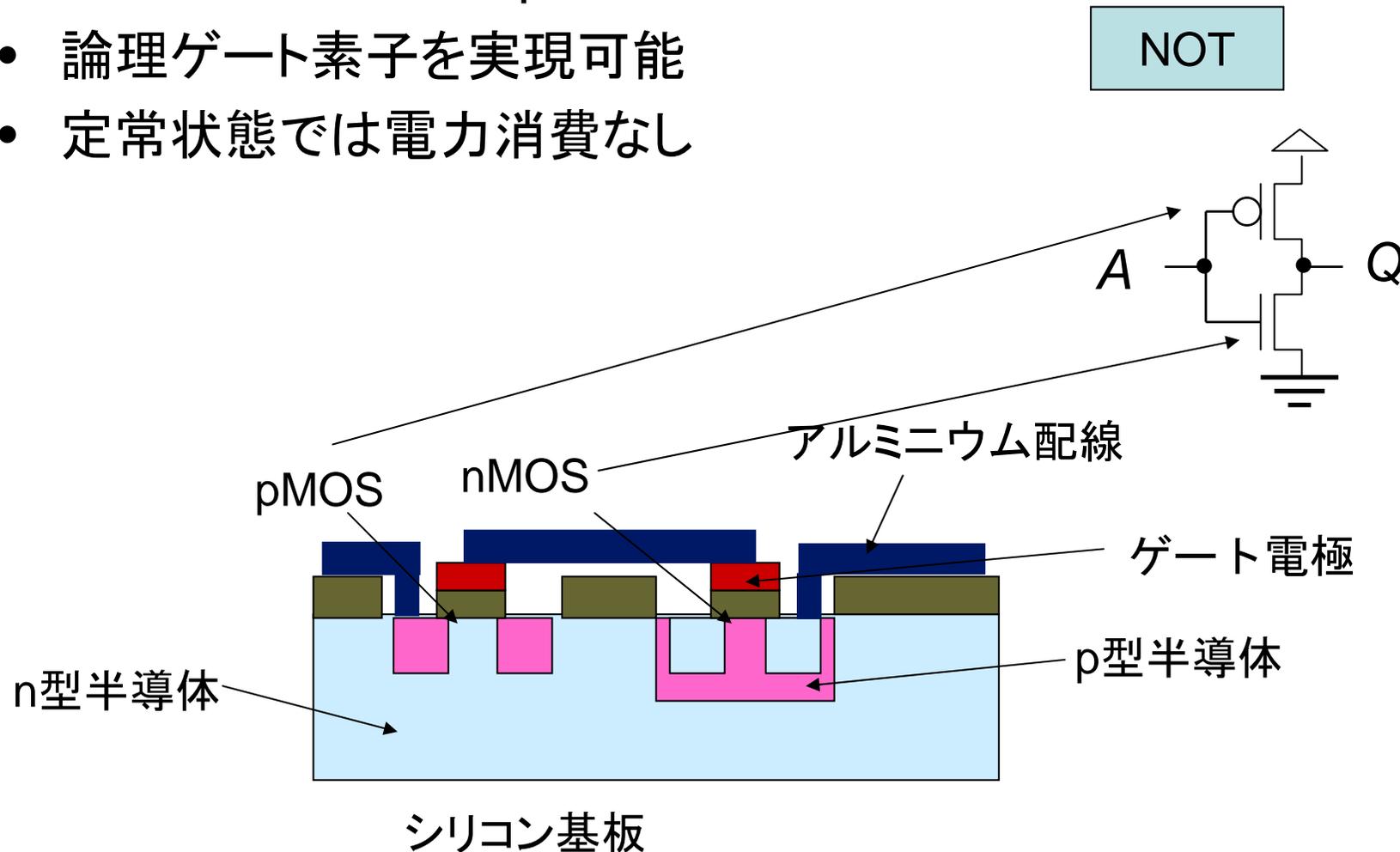
- 近年のコンピュータは半導体素子の中でもトランジスタを利用
 - かつては真空管とかリレーとか
- さらに、トランジスタを1つのシリコンチップに集積した**集積回路(IC: Integrated Circuit)**を利用

(1) 集積回路

- バイポーラ型
 - バイポーラ型トランジスタを組み合わせた集積回路
 - 回路の動作が高速
 - 消費電力が大きく、発熱量が多くなる
 - 90年代中盤に集積回路の主流から外れたが、強い電流を出力できるという利点があるので、使われている分野もある
- CMOS(Complementary Metal Oxide Semiconductor)型
 - 2種類のMOSFETを組み合わせた集積回路
 - 消費電力が少ない
 - バイポーラ型よりも集積できるので安価に製造可能
 - 現在のコンピュータの多くはCMOS型集積回路によって構成

シリコンチップ上のCMOS

- CMOSとはnMOSとpMOSの相補構成
- 論理ゲート素子を実現可能
- 定常状態では電力消費なし



(2) ICメモリ

- 大きくROMとRAMに分けられる
 - ROM(Read Only memory): **不揮発性**
 - 電源を切ってもデータは残る
 - RAM(Random Access Memory): **揮発性**
 - 電源を切ったらデータは消失する
- 上記のRAMとROMの両方の性能を満たすことを目的とした物も開発中
 - Magnetic RAM
 - Phase Change RAM
 - Registive RAM

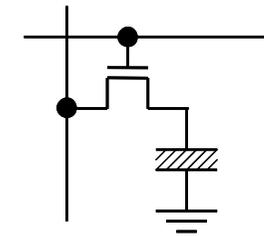
ROM(Read Only Memory)

- マスクROM
 - 製造時に記憶内容を書き込み、書き換えることはできない
- PROM(Programmable ROM)
 - 一度だけデータを書き込めるROM。書き換えることはできない
- EPROM(Erasable PROM)
 - 記憶内容を紫外線で一括消去して、内容を書き換えることができる
- EEPROM(Electrically EPROM)
 - 電氣的にブロックまたはバイト単位で記憶内容を消去し、データを書き換えることができる
- フラッシュメモリ
 - 電氣的に一括またはブロック単位でデータを消去して内容を書き換えることができる
 - SDカードなどの記録メディアに主に使われる
 - 近年では、SSDという形でHDDの代替に使われることもある

RAM(Random Access Memory)

- DRAM(Dynamic RAM)

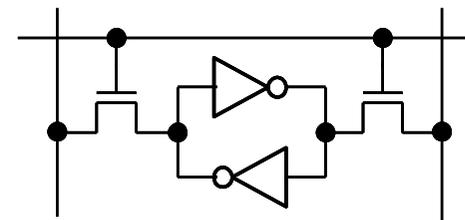
- コンデンサを使用してデータを記憶する
- コンデンサ中の電荷は漏れるため、一定時間ごとに再書き込み(リフレッシュ)を行い、記憶内容を保持
- 集積化しやすく、低コスト化、大容量化に対応
 - >主記憶装置に用いられる。



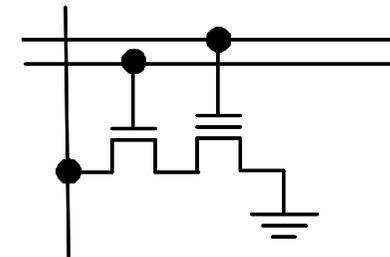
DRAM

- SRAM(Static RAM)

- フリップフロップ回路を用いてデータを記憶する
- DRAMに比べて高速
- キャッシュメモリに用いられる



SRAM



フラッシュ
メモリ

3.3 プロセッサアーキテクチャ

- 機械語
 - 0/1の羅列 ->人間にはまずちんぷんかんぷん
 - CPUの命令セットが違えば異なる
- アセンブリ言語
 - 機械語と1対1に対応する表記
 - 人間にはだいぶ分かりやすい
 - アセンブラで機械語に変換可能
- 高級言語(例: C言語)
 - 人間に分かりやすい言語
 - コンパイラでアセンブリ言語に変換可能

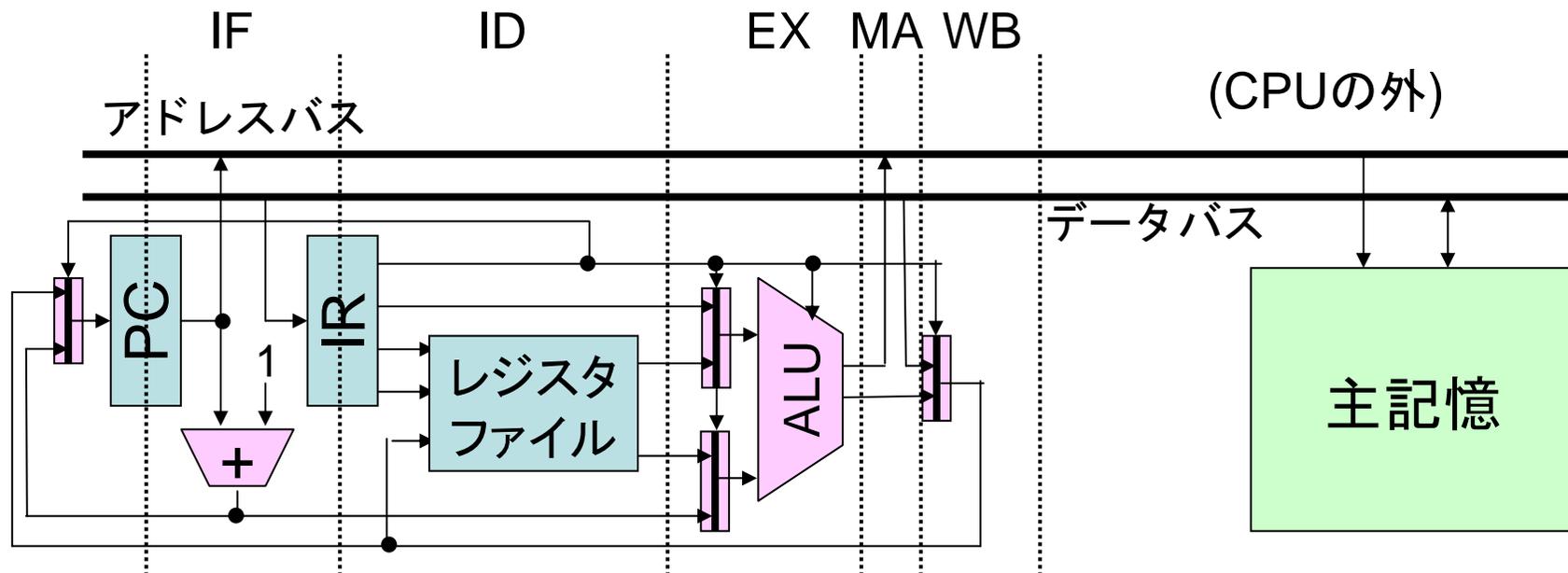
•C言語 コンパイラ •アセンブリ言語 アセンブラ •機械語
c = a + b  add r3, r1, r2  0101100011010001

CPUに関するアーキテクチャ

- アーキテクチャ: 構造、建築術
- CPUに関するアーキテクチャ
 - 命令セットアーキテクチャ: 機械語の構造
 - ISA: Instruction Set Architecture
 - マイクロアーキテクチャ: 回路の構造
 - ロード/ストアアーキテクチャ: 主記憶の内容は一度レジスタに読み込まないと演算できない構造
- ISAが同じでマイクロアーキテクチャが違うものは多数ある
 - Intel社Core-i7とAMD社Phenom
 - ARM-v7 ISAのCPU(携帯電話等で多用)は非常に多数ある

(2) CPUにおける命令の実行

- 以下の5段階(フェーズ)で命令を実行
 - 命令フェッチ(IF: Instruction Fetch)
 - 命令デコード(ID: Instruction Decode)
 - 実行(EX: EXecution)
 - メモリ(主記憶)アクセス(MA: Memory Access)
 - ライトバック(WB: Write Back):

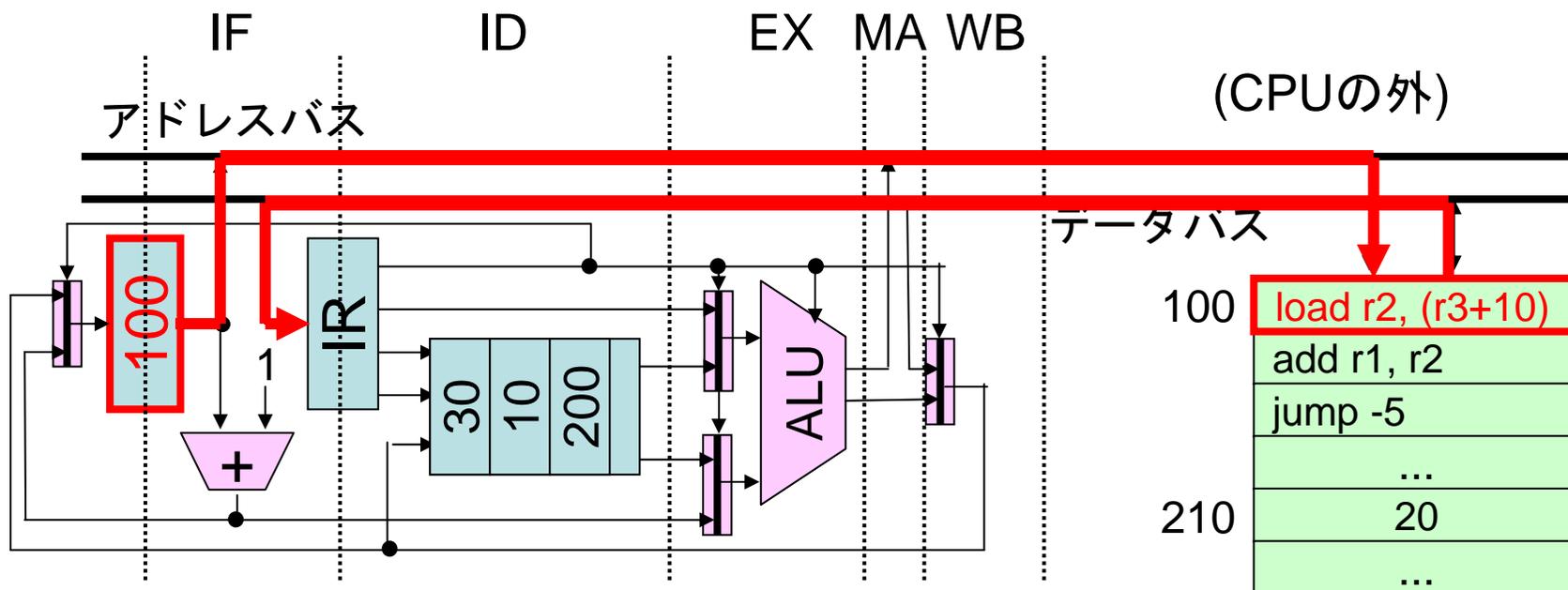


3つの命令による命令実行の例示

- メモリアクセス命令(ロード／ストア命令)
 - メモリからレジスタに値を読み込むロード命令の例示
 - `load r2, (r3+10)`
 - レジスタ2番に、メモリ上の(レジスタ3番の値+10)番地の値を読み込む
- 演算命令
 - レジスタ値同士を加算する加算命令の例示
 - `add r1, r2`
 - レジスタ1番とレジスタ2番の値を加算し、結果をレジスタ1番に格納
- 制御命令(分岐命令)
 - 無条件に分岐するジャンプ命令の例示
 - `jump -5`
 - PCを-5し、5つ前の命令の実行に戻る

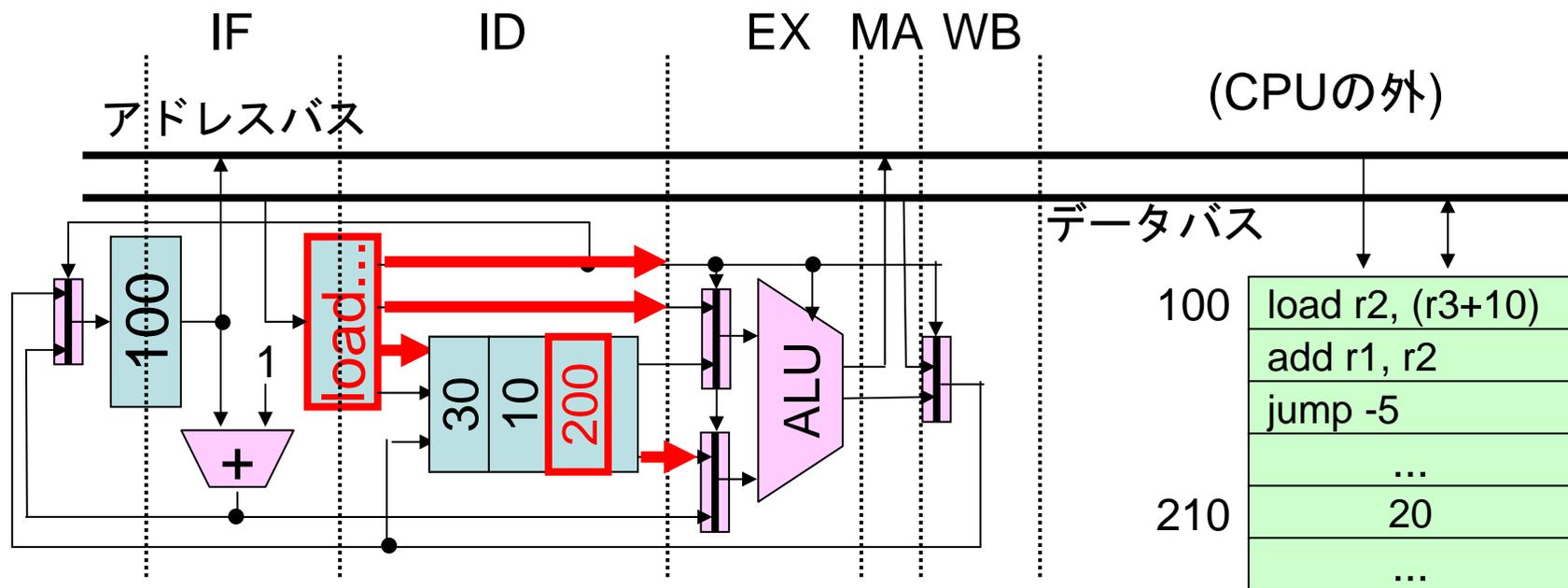
ロード命令の実行(IF)

- IF: Instruction Fetch
- PC(Program Counter)の値を主記憶に送り、対応する番地の命令語を読み出す
- 命令語はIR(Instruction Register)に保存



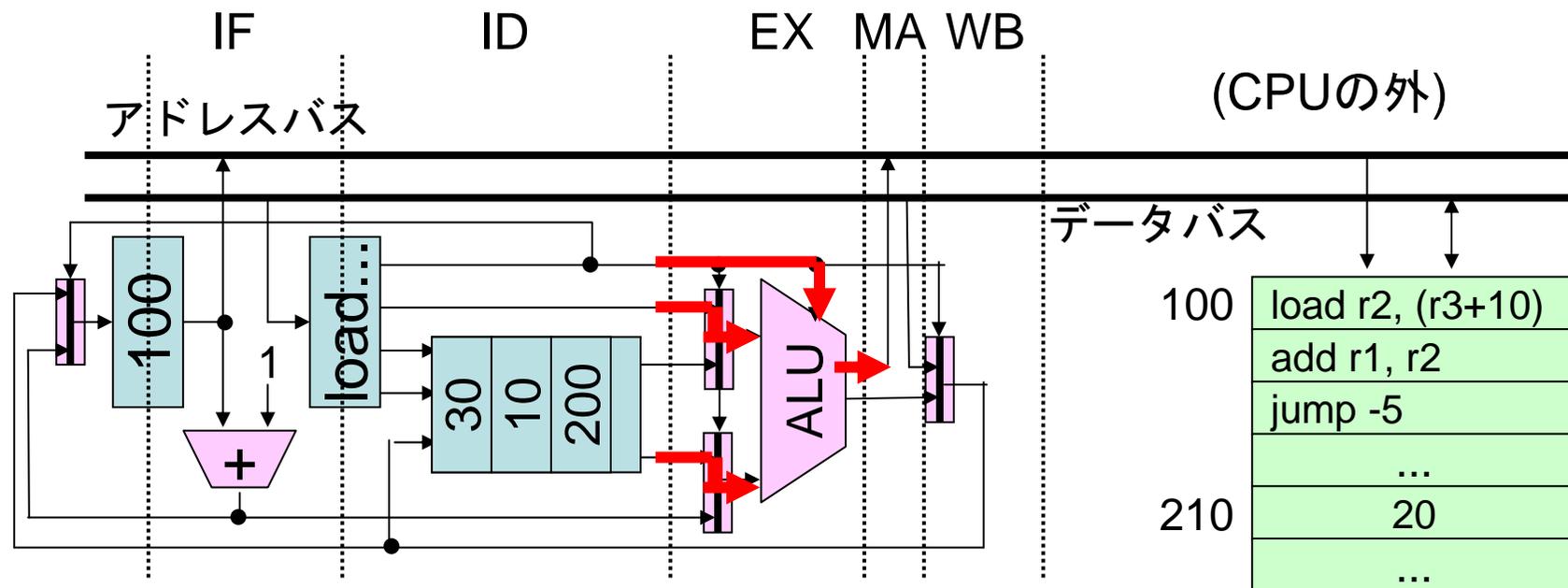
ロード命令の実行(ID)

- ID: Instruction Decode
- レジスタファイルからレジスタ値r3の読み出し
- 命令語から即値10の切り出し
- 命令語から演算の内容等を切り出し



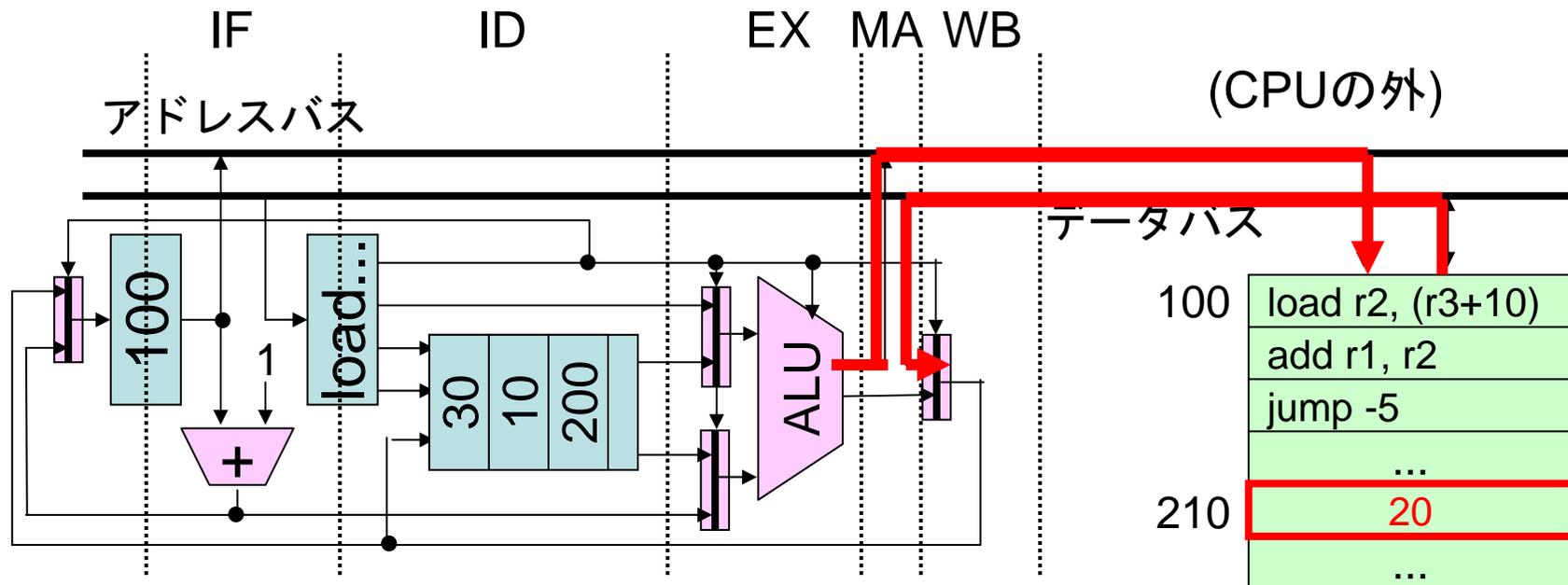
ロード命令の実行(EX)

- EX: EXecution (EXEとも略す)
- ALU(Arithmetic Logic Unit)で実効アドレスを生成
 - レジスタr3の200と即値の10を加算



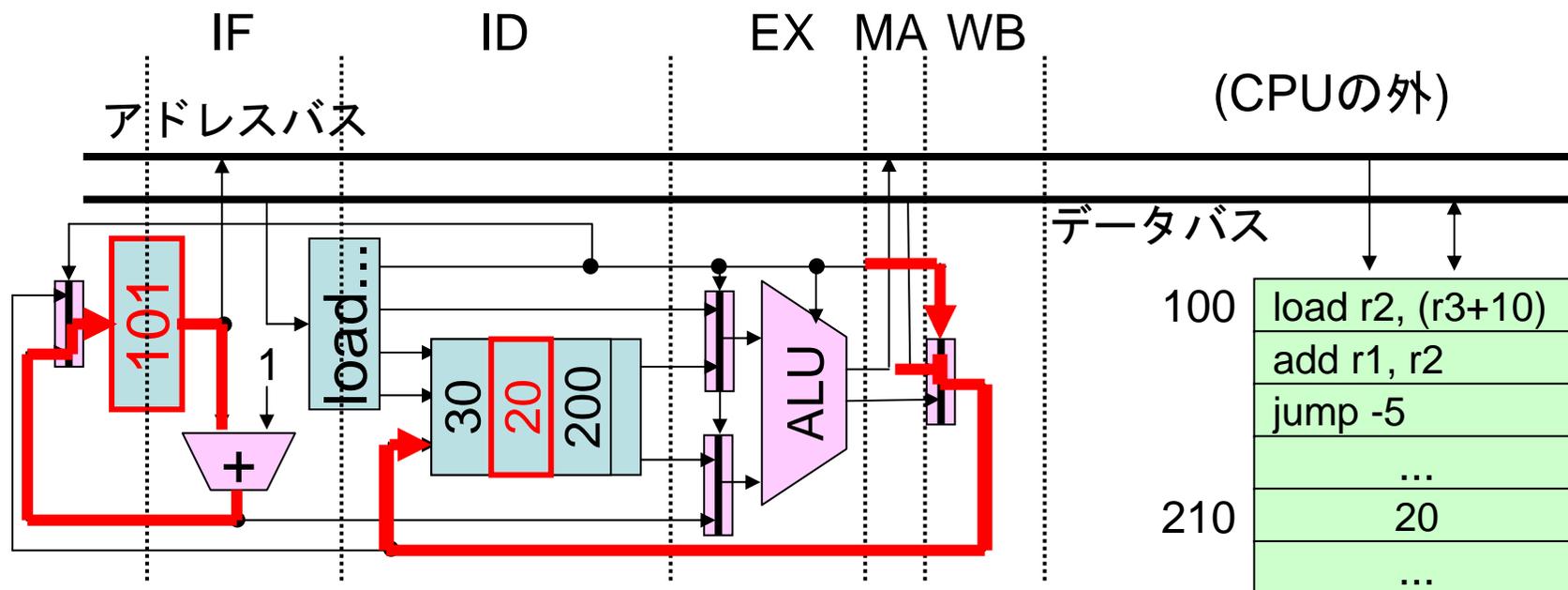
ロード命令の実行(MA)

- MA: Memory Access (MEMとも略す)
- 計算した実効アドレスを主記憶に送り、データを読み出す



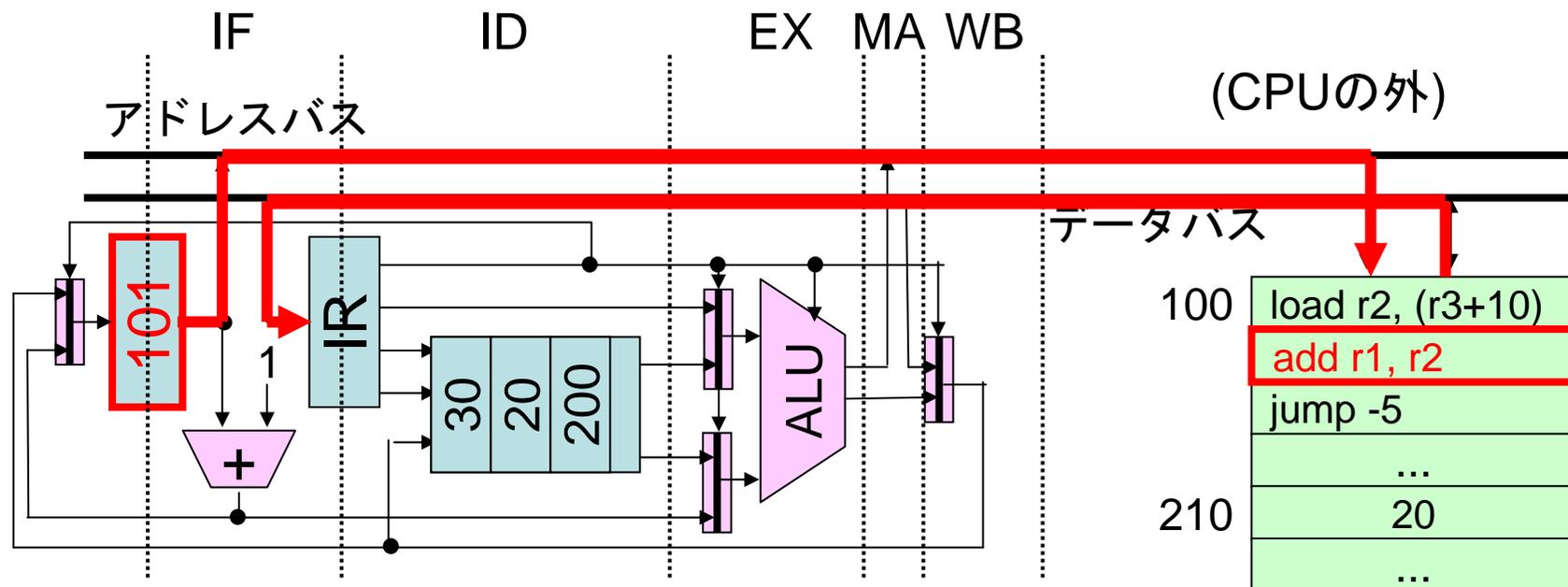
ロード命令の実行(WB)

- WB: Write Back
- ロードした結果をレジスタr2として保存
- 次の命令の実行に備え、PCを+1して次の命令を指し示す
 - IFの直後に+1するマイクロアーキテクチャも多い



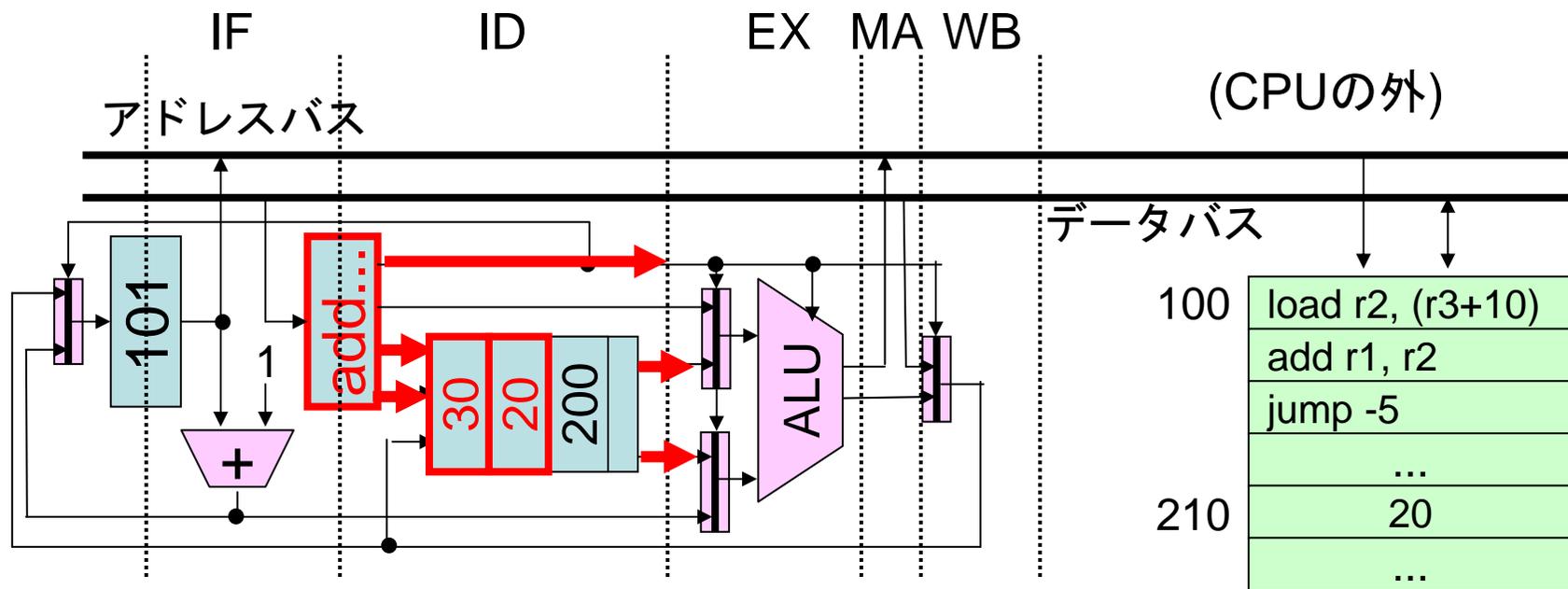
演算命令の実行(IF)

- ロード命令と変わらず



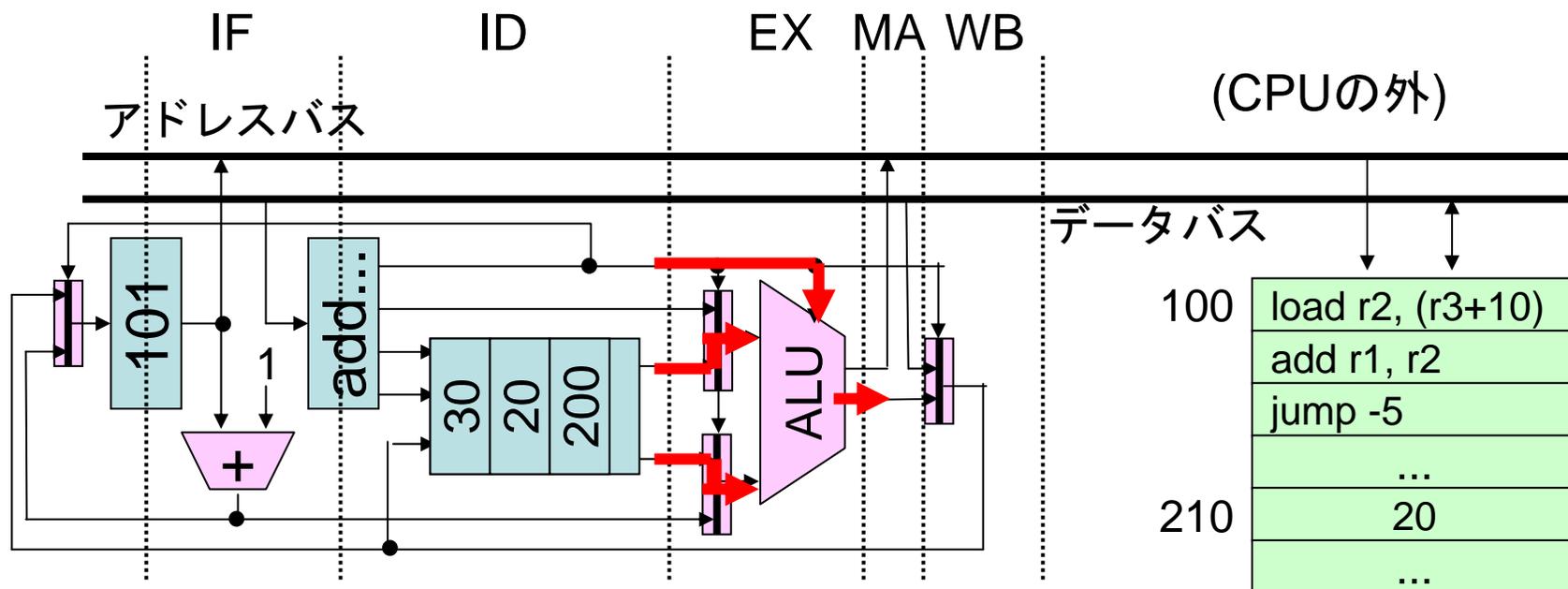
演算命令の実行(ID)

- レジスタファイルから2個のレジスタ値(r1, r2)を読み出す
- 命令語から演算の内容等を切り出し



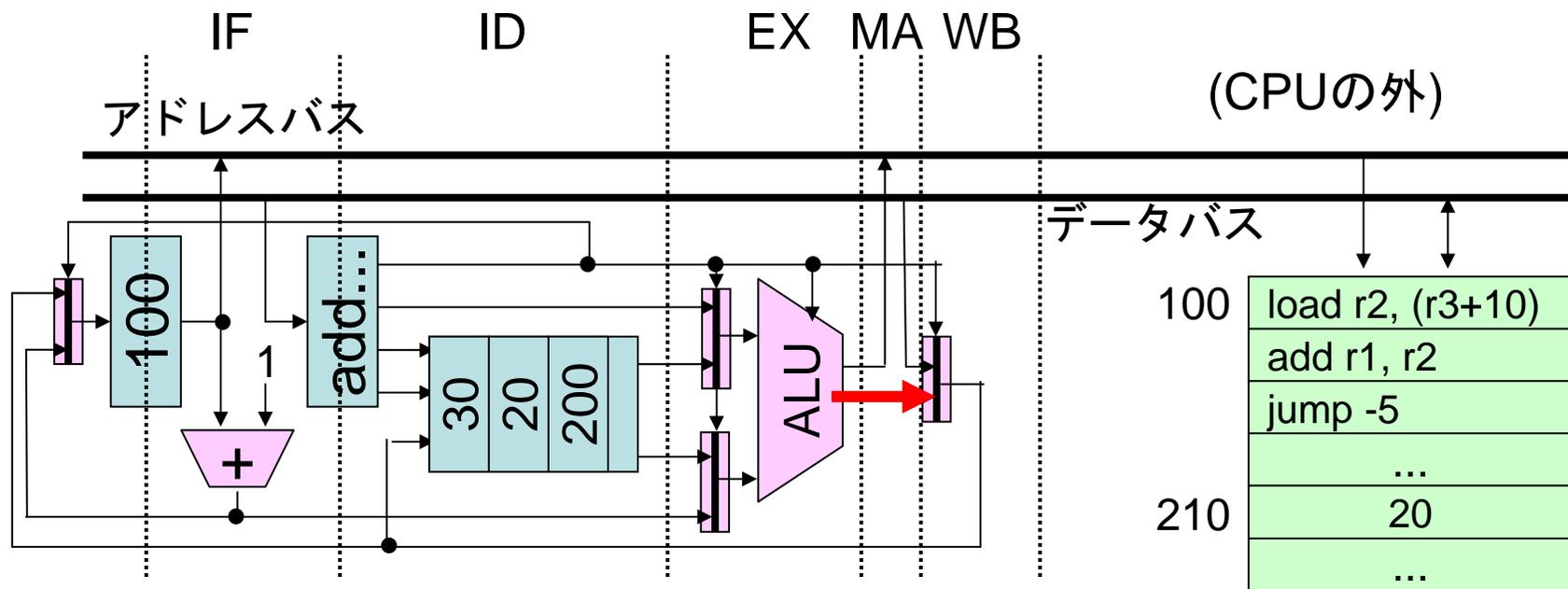
演算命令の実行(EX)

- ALUで2つのレジスタ値を加算
 - レジスタr1の30とレジスタr2の20を加算



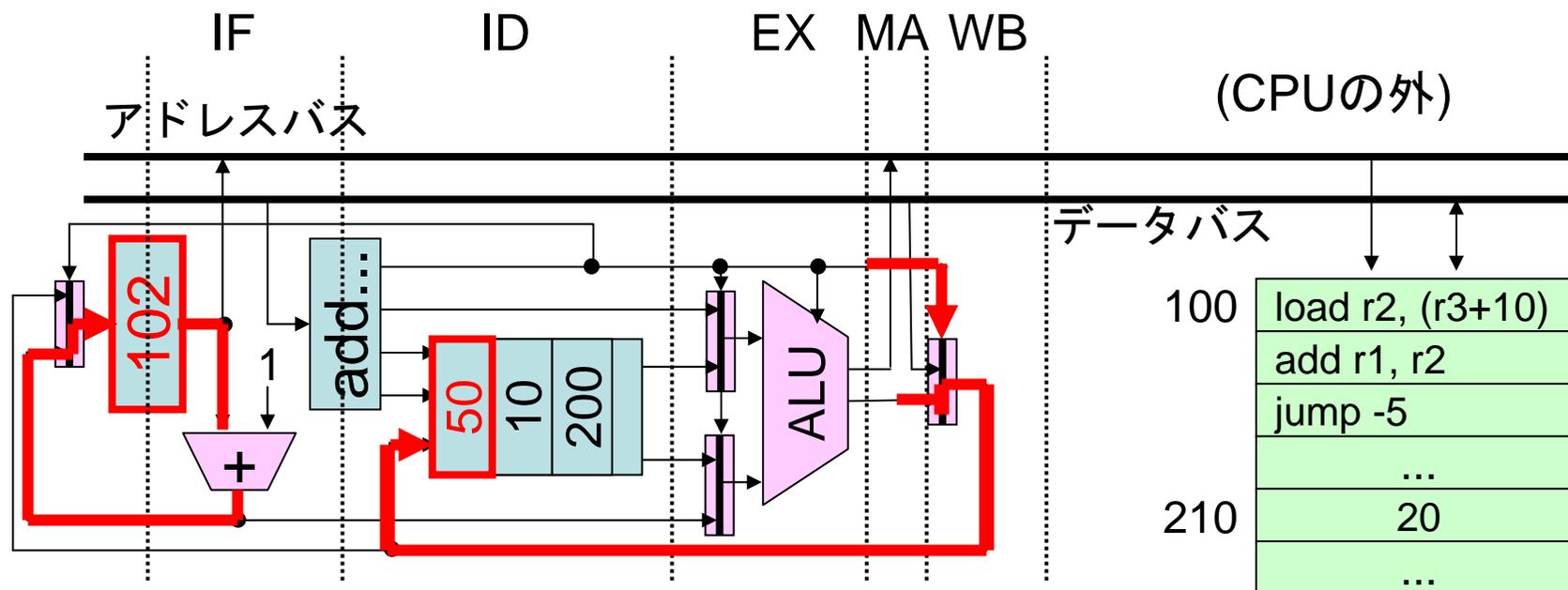
演算命令の実行(MA)

- 何も行わない
 - 主記憶アクセスがない命令は全て同じ



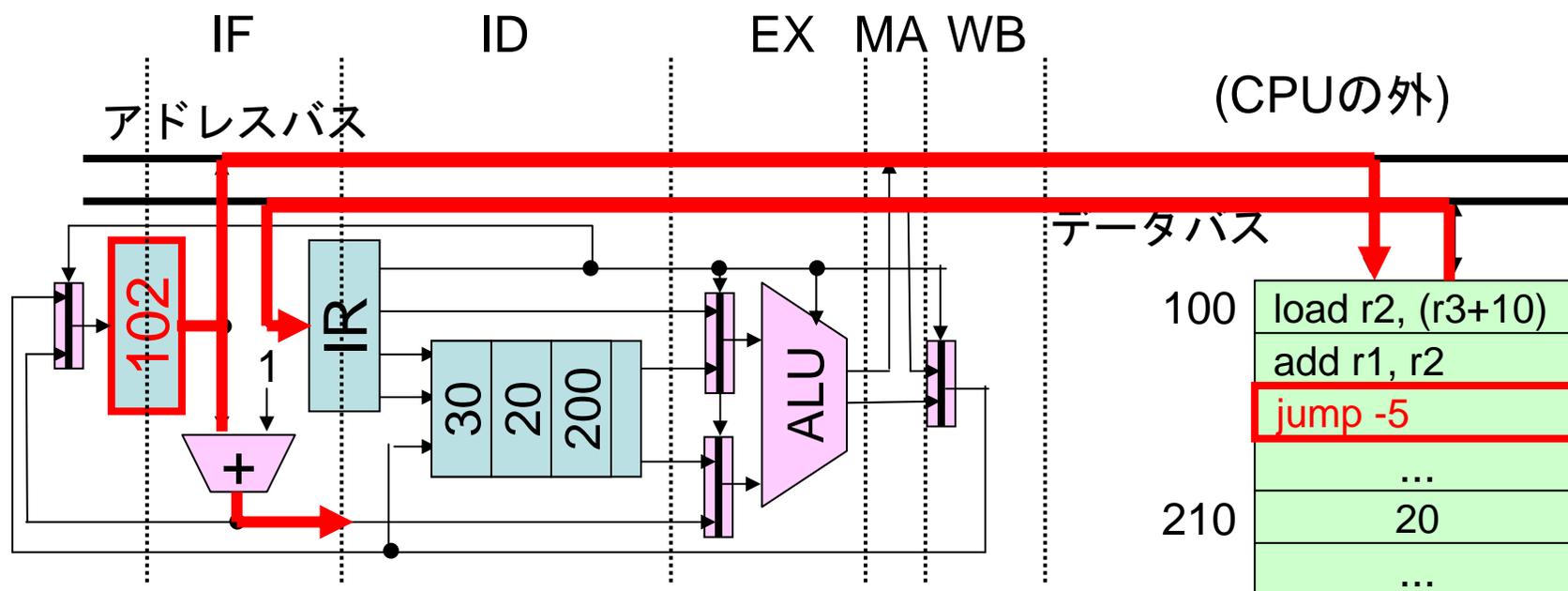
演算命令の実行(WB)

- 演算結果をレジスタr1として保存
- 次の命令の実行に備え、PCを+1して次の命令を指示示す



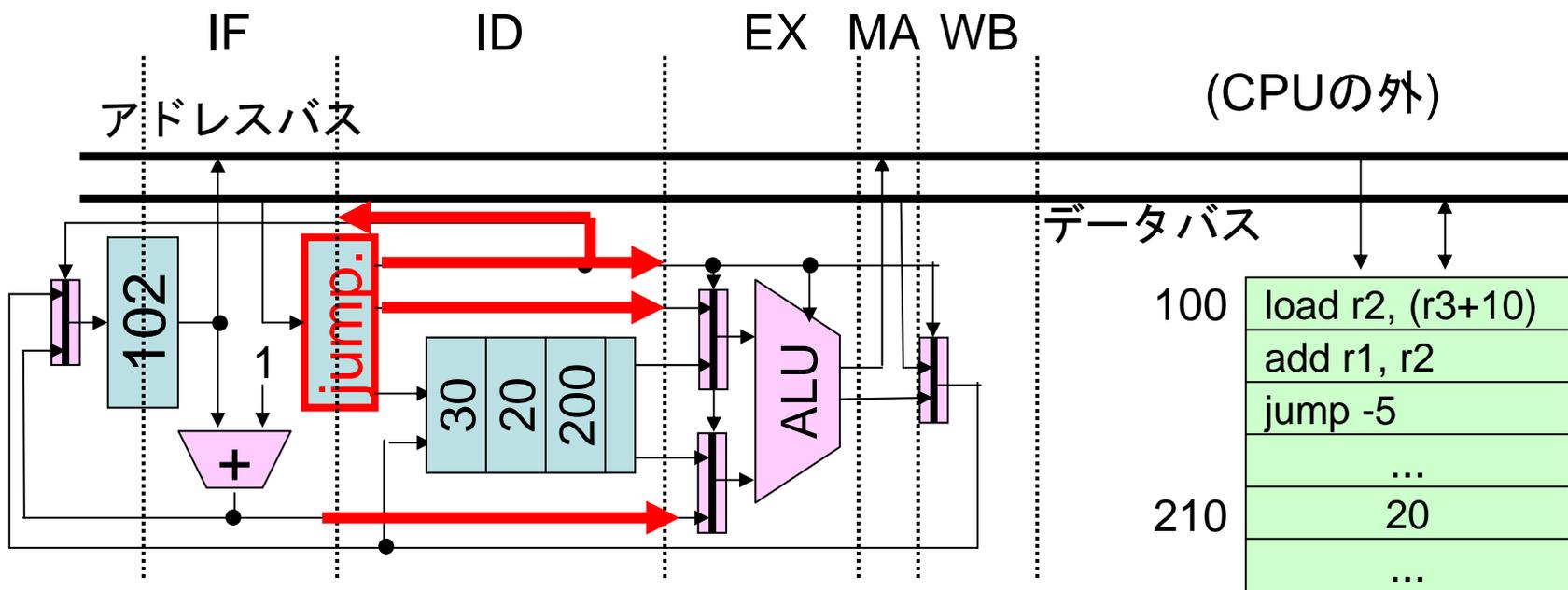
ジャンプ命令の実行(IF)

- ロード命令、加算命令と同じ
- あえて言うならば、PC+1の値が分岐先の計算の為にALUに送られる



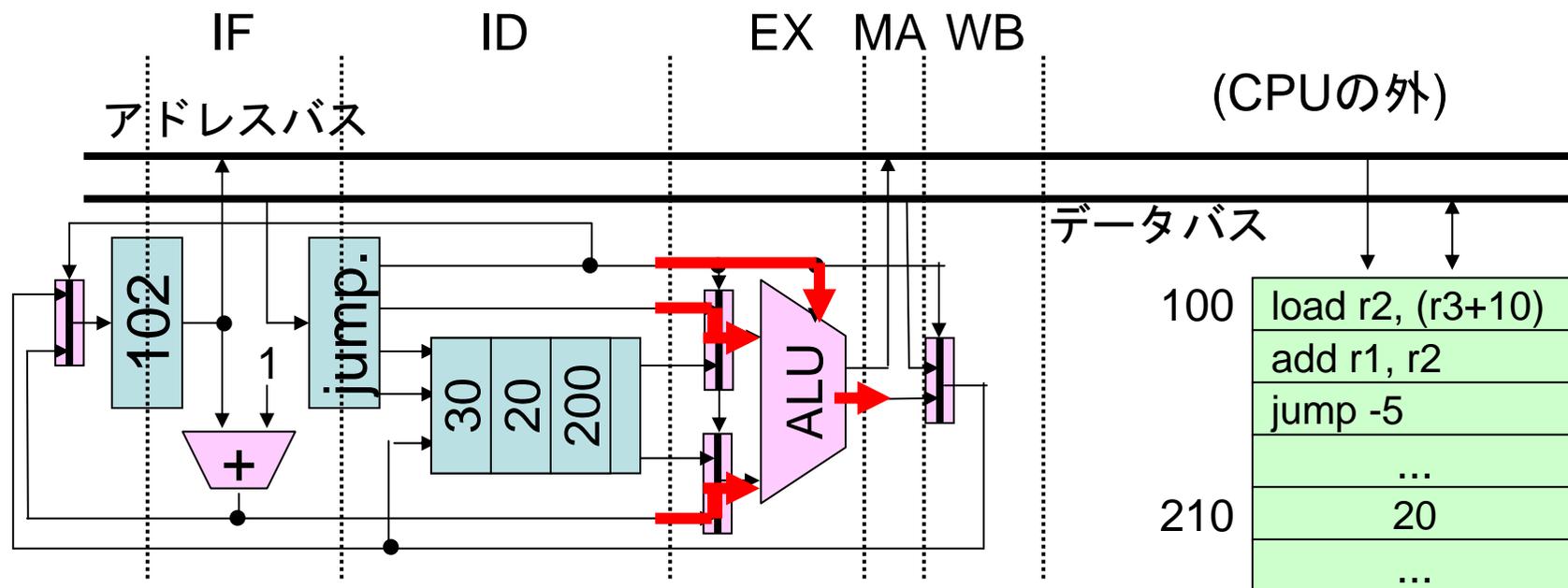
ジャンプ命令の実行(ID)

- PC+1の値をALUに送る
- 命令語から即値-5を切り出す
- PCの更新をALUの出力に切り替える準備



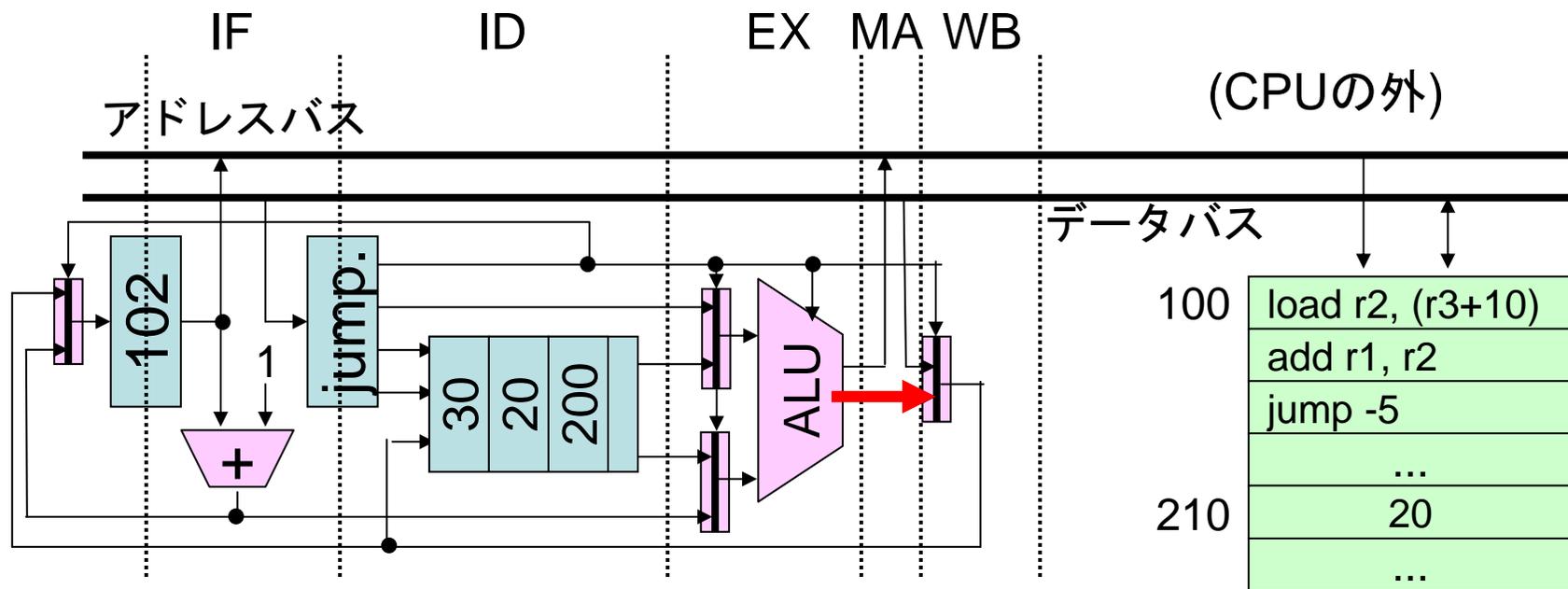
ジャンプ命令の実行(EX)

- PC+1(103)と即値-5を加算する



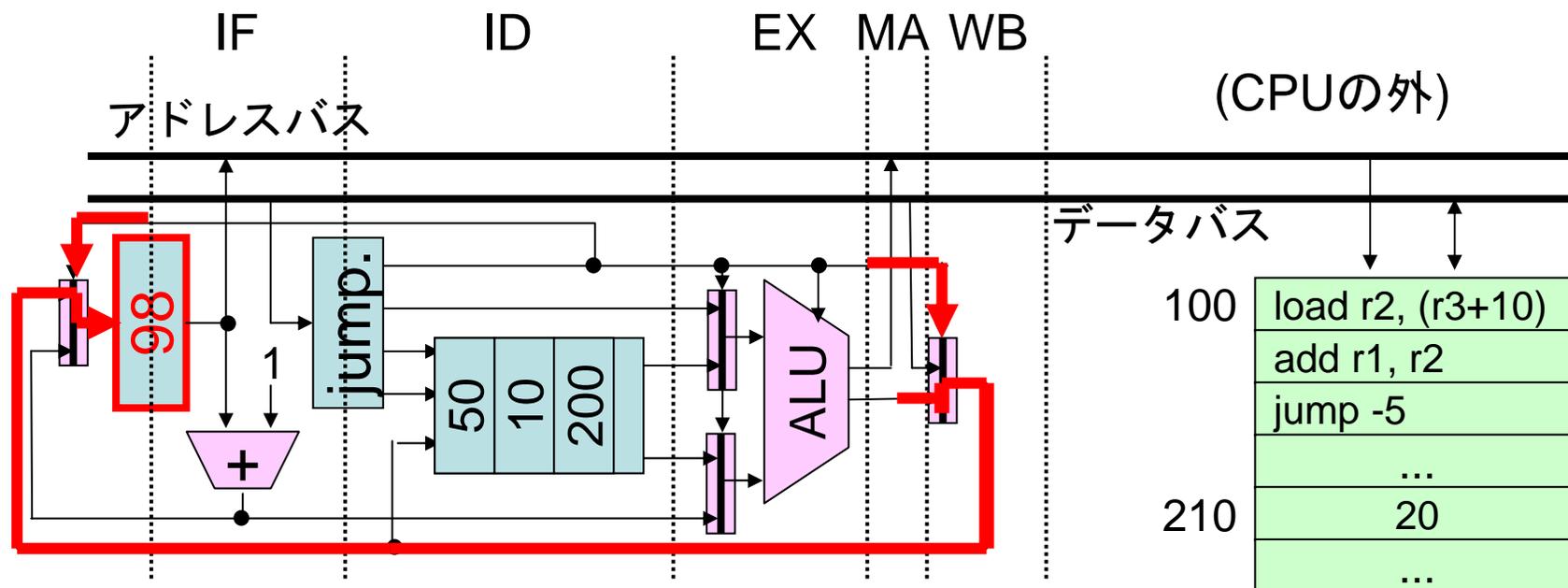
ジャンプ命令の実行(MA)

- 何もしない



ジャンプ命令の実行(WB)

- PCの値を演算結果である98に更新

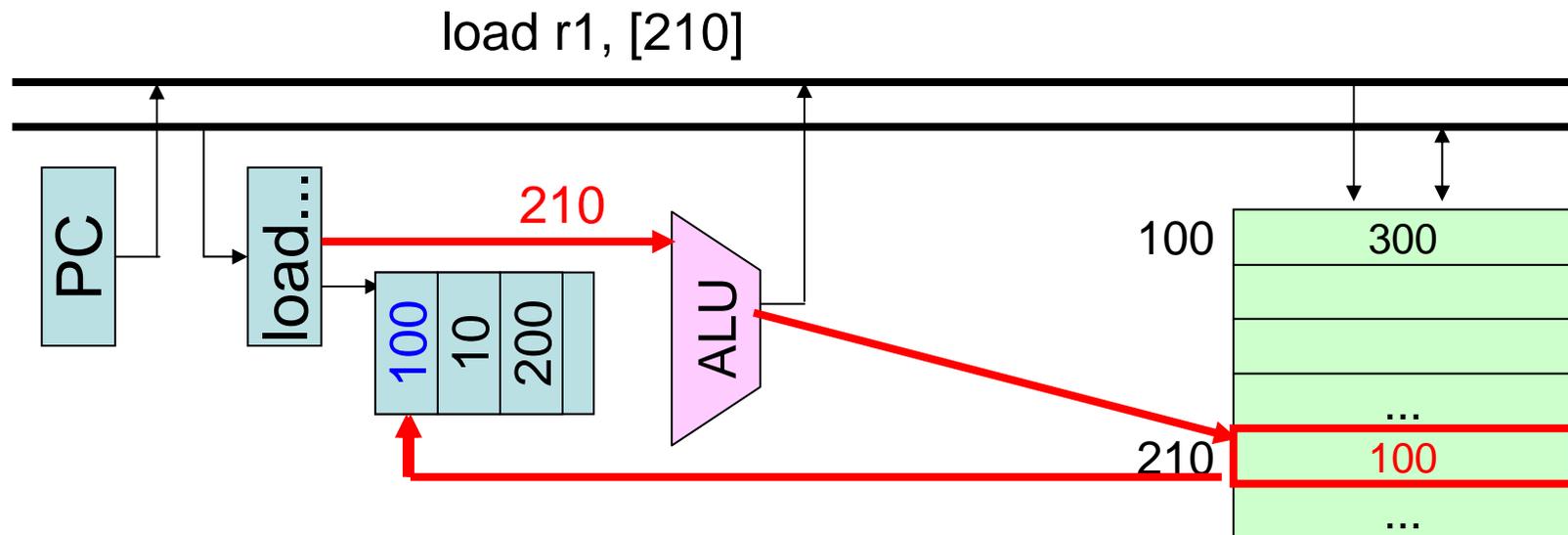


(3) アドレス指定方式

- メモリ中の命令語やデータは**実効アドレス**で示す
 - 様々なアドレス指定方式(**アドレッシングモード**)が存在
 - 直接アドレス指定方式
 - オフセット付きレジスタ間接方式
 - メモリ間接指定方式
 - 即値オペランド指定方式
- うまく使い分けるとプログラムの記述が容易に

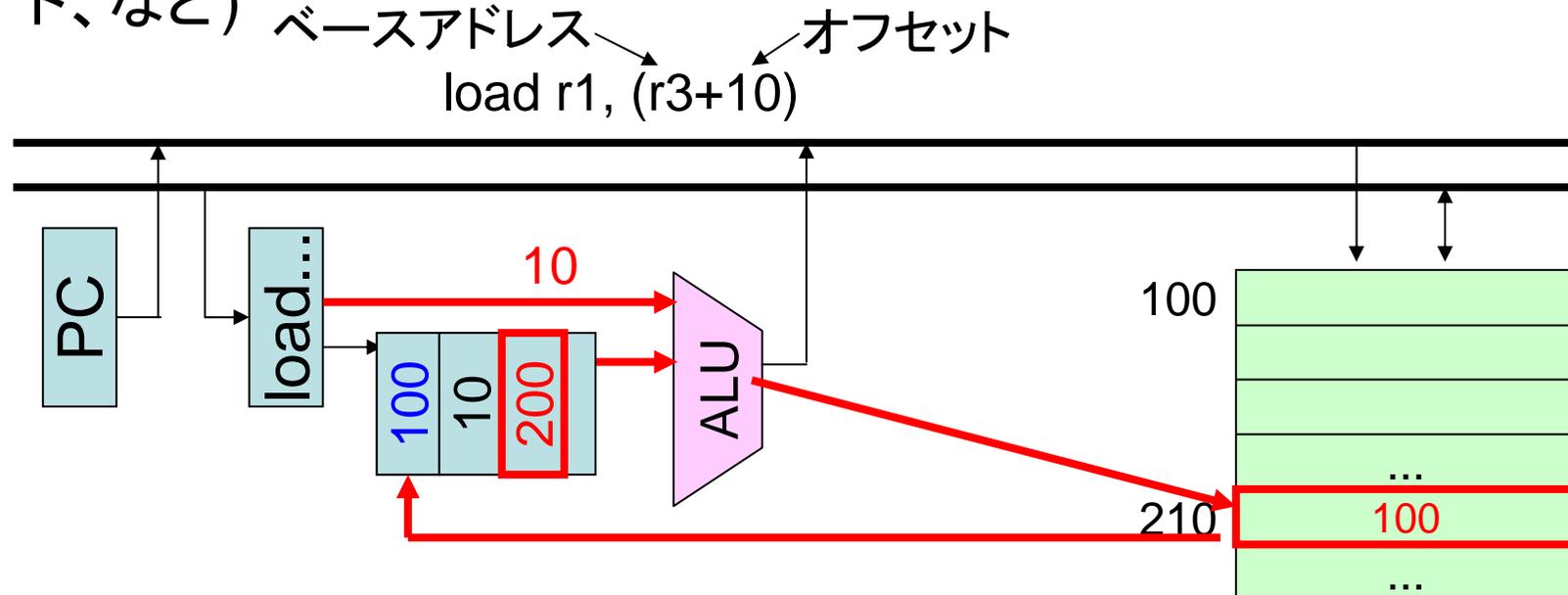
(a) 直接アドレス指定方式

- 命令語の一部をアドレスにする方法
- 命令語の長さに制約される



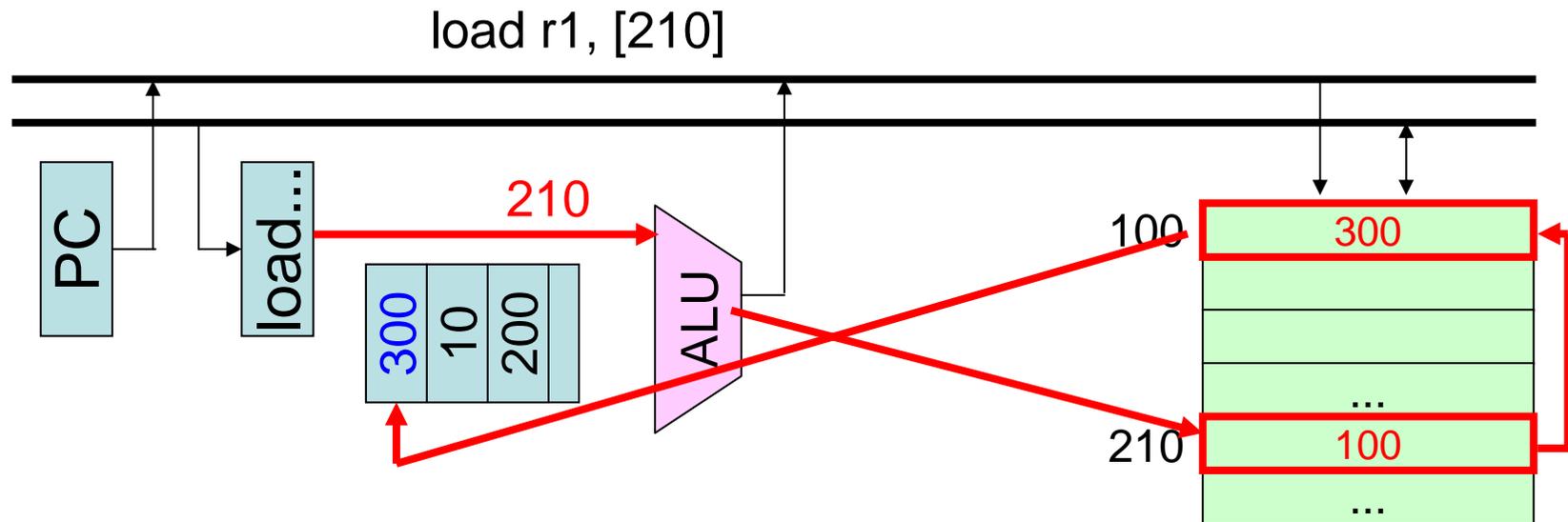
(b) オフセット付きレジスタ間接方式

- もっともよく使われる方式
- 配列のアクセス等で便利
 - 変数nをベースアドレスに
 - 変数“n+1”の+1をオフセットに
- 派生も多い(例: ベースのみ、ベース+インデクス+オフセット、など)



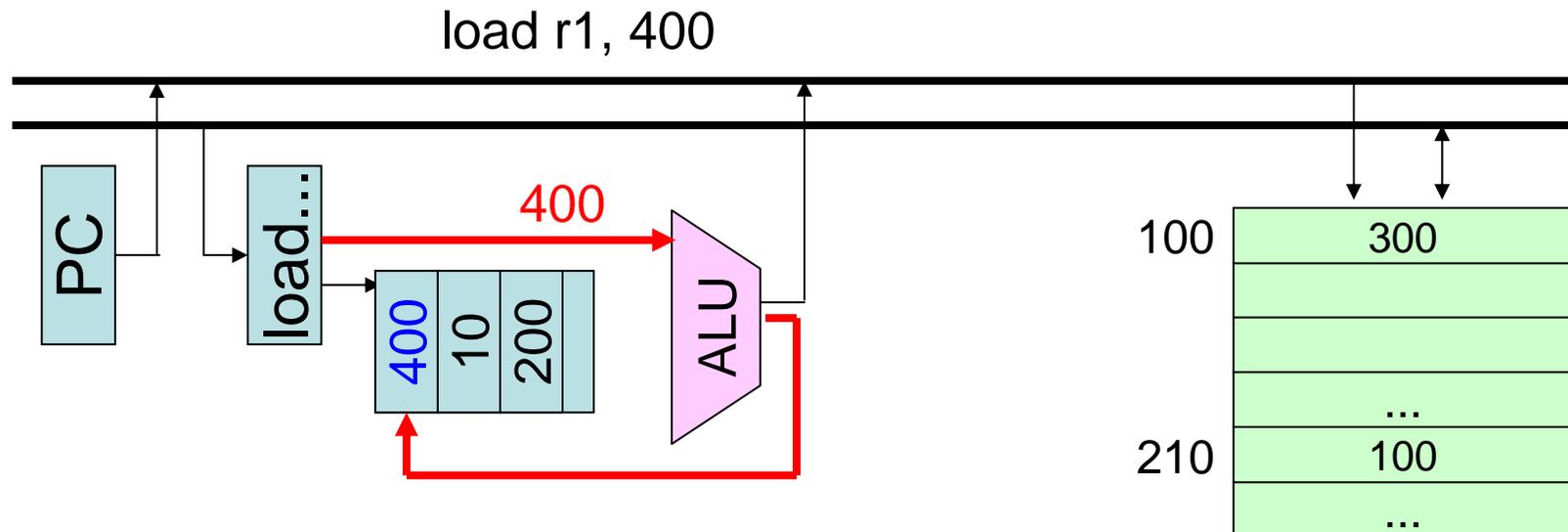
(c) メモリ間接指定方式

- 主記憶内のデータが実効アドレスとなる方式
 - 最初の実効アドレスは別途生成
- ライブラリ関数の呼び出し時のジャンプテーブルなどに便利
- 動作複雑なので使えないISAも多い



(d) 即値オペランド指定方式

- 命令語の一部を直接レジスタ値にする
- 主記憶アクセスは行われない



(4) CPUの性能指標

- クロック周波数
 - CPUは、コンピュータ内部にある発振器から発せられる信号に同期して動作
 - 発振器が「一秒間に何回発振しているか」という指標が、クロック周波数
 - クロック周波数が1GHz=1秒間に10億回の信号の振幅が発生
 - 同じ設計ならば、クロック周波数が高い方が性能が高くなる
- CPI (Clock cycles Per Instruction)
 - 1命令を実行するのに要するクロックサイクル数(クロックの振幅の数)
 - 例: Intel社のPentium 4の場合、1命令の実行に4~30クロックサイクル程度を必要
 - 命令によって、CPIは異なる
 - 後述するパイプライン化では、処理が並列に行われているため、平均CPIは単一命令の実行時よりも小さくなる

(4) CPUの性能指標

- MIPS(Million Instructions Per Second)
 - 1秒間に実行できる命令の数を100万単位で表したものの
 - 25MIPS = 25,000,000命令を1秒間に実行
- FLOPS(FLoting oint Operations Per Second)
 - 1秒間に実行できる浮動小数点演算(倍精度浮動小数点)の数
 - スーパーコンピュータの性能指標によく使われる
 - 例: 京コンピュータは10PFLOPSを目指して設計

ピーク性能(理論最大値)と
実効性能(プログラムを動作させて測定した値)に注意

処理性能の計算例

- 1.0×10^{15} 個の行列演算を行う科学技術計算を 1.0×10^{12} FLOPS(1TFLOPS)のスーパーコンピュータで実行する時の実行時間は？
 - ただし、演算に付随する処理は取るに足らないほど短時間で終了すると考える

$$1.0 \times 10^{15} \div 10 \times 10^{12} = 1000 \text{秒}$$

処理性能の計算例

- 2GHzで動作しているCPUであるプログラムを動作させた時の平均CPIが0.5の場合、1秒あたりに実行される命令数は？

$$2.0 \times 10^9 \div 0.5 = 4 \times 10^9 \text{命令} / \text{秒}$$

処理性能の計算例

- 以下のような800MHzで動作するCPUの1秒あたりに実行される命令数は？
 - 平均CPIが演算命令で1、メモリアクセス命令で5、分岐命令で10である
 - 演算命令50%、メモリアクセス命令30%、分岐命令20%のプログラムを実行した時

$$\begin{aligned} \text{平均CPI} &= 1 \times 0.5 + 5 \times 0.3 + 10 \times 0.2 = 4 \\ 800 \times 10^6 \div 4 &= 200 \times 10^6 \text{命令/秒} \end{aligned}$$

演習

- 3GHzで動作するCPUの平均CPIが浮動小数点演算命令で3、メモリアクセス命令で10、分岐命令で15である。浮動小数点演算命令50%、メモリアクセス命令40%、分岐命令10%のプログラムを実行した時の1秒あたりに実行される命令数は？

平均CPIはそれぞれのCPIと出現率の積で求まる。

$$\begin{aligned}\text{平均CPI} &= 3 \times 0.5 + 10 \times 0.4 + 15 \times 0.1 \\ &= 1.5 + 4 + 1.5 = 7\end{aligned}$$

クロック周波数を平均CPIで割れば1秒あたりの実行命令数は求まる。

$$\text{1秒あたりの実行命令数} = 3 \times 10^9 / 7 \doteq 428 \times 10^6$$

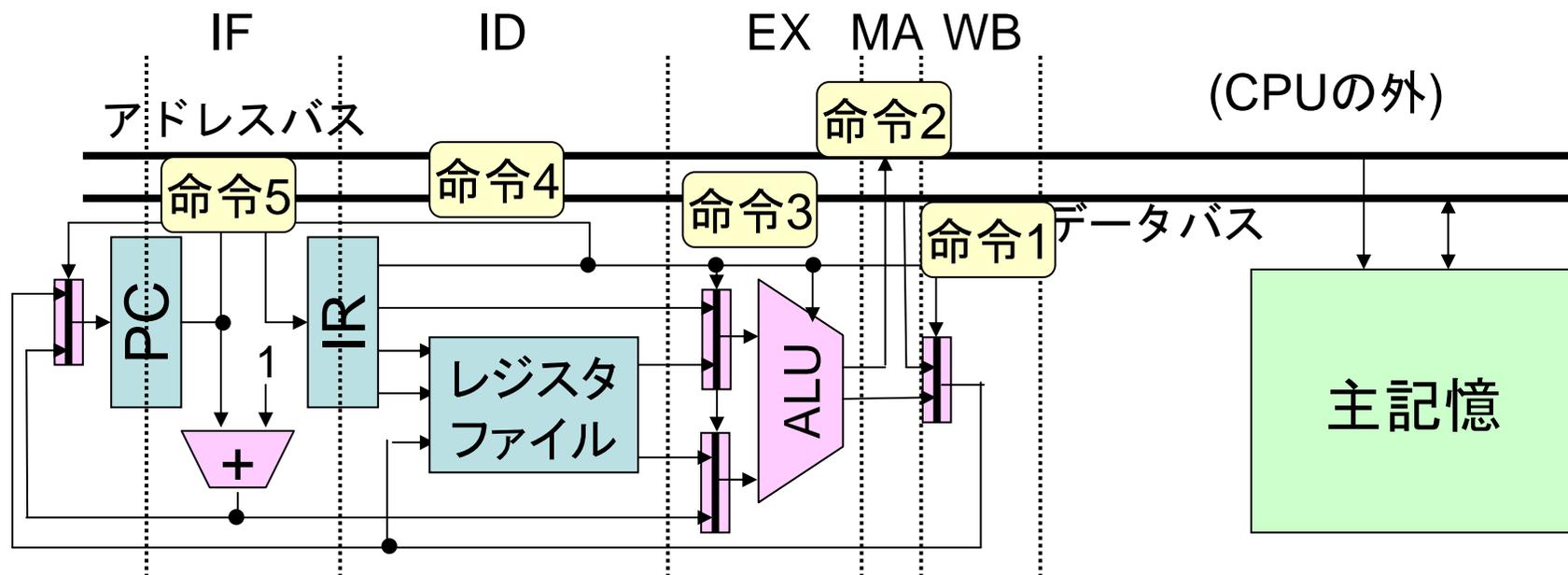
(5) CPUの高速化技術

- CPUが高速化するとあらゆる処理が早くなる
->様々な高速化技術が提案されている
- 紹介する高速化技術
 - パイプライン処理
 - RISCとCISC
 - スーパスカラ
 - VLIW(Very Long Instruction Word)
 - マルチプロセッサ

(a) パイプライン処理

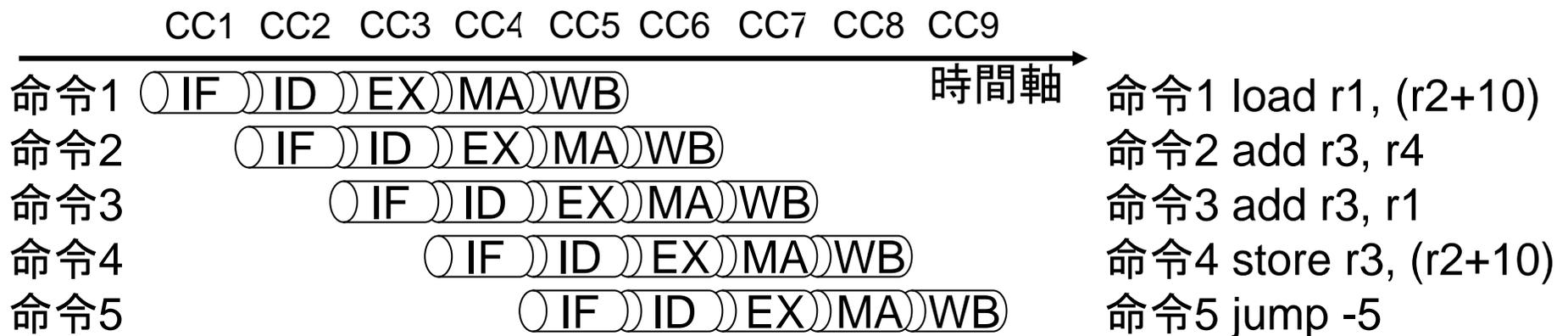
- ベルトコンベア上の流れ作業をイメージ
- 現在はさらに細かく作業を分割
 - スーパーパイプライン
- 流れ作業が停止することもある
 - **パイプラインストール**と呼ぶ
 - 原因はXXハザードと名づけられている

命令1 load r1, (r2+10)
命令2 add r3, r4
命令3 add r3, r1
命令4 store r3, (r2+10)
命令5 jump -5



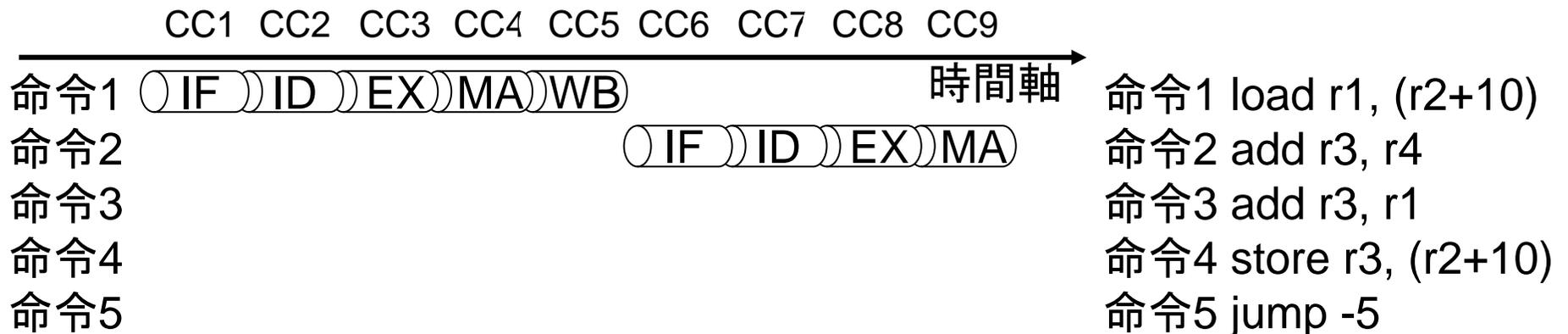
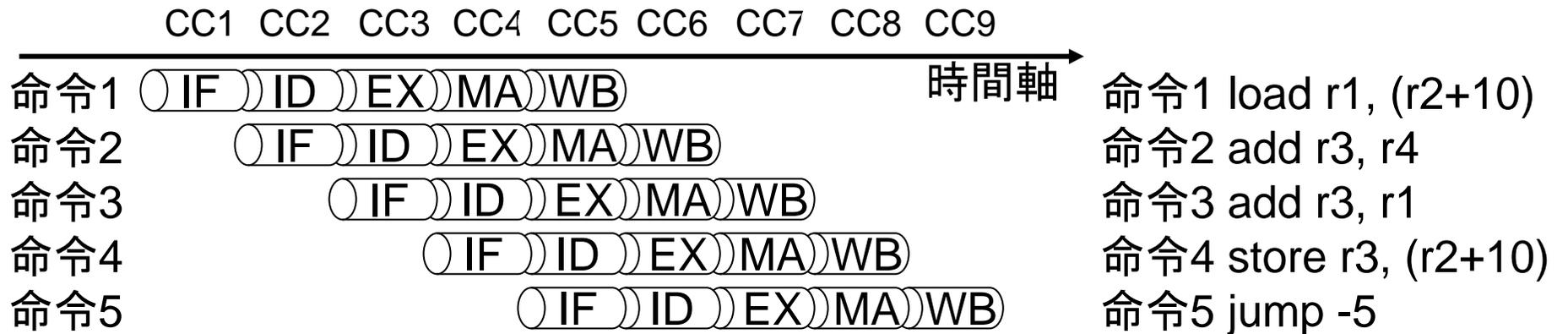
パイプライン処理の時間軸表記

- 下記のような時間軸を用いた表現も行う
 - 時間の単位は**クロックサイクル**(CC)
 - 同一クロックサイクルの処理は上下方向で表される
 - 前の図はCC5の状態



パイプライン化なしとの時間軸の比較

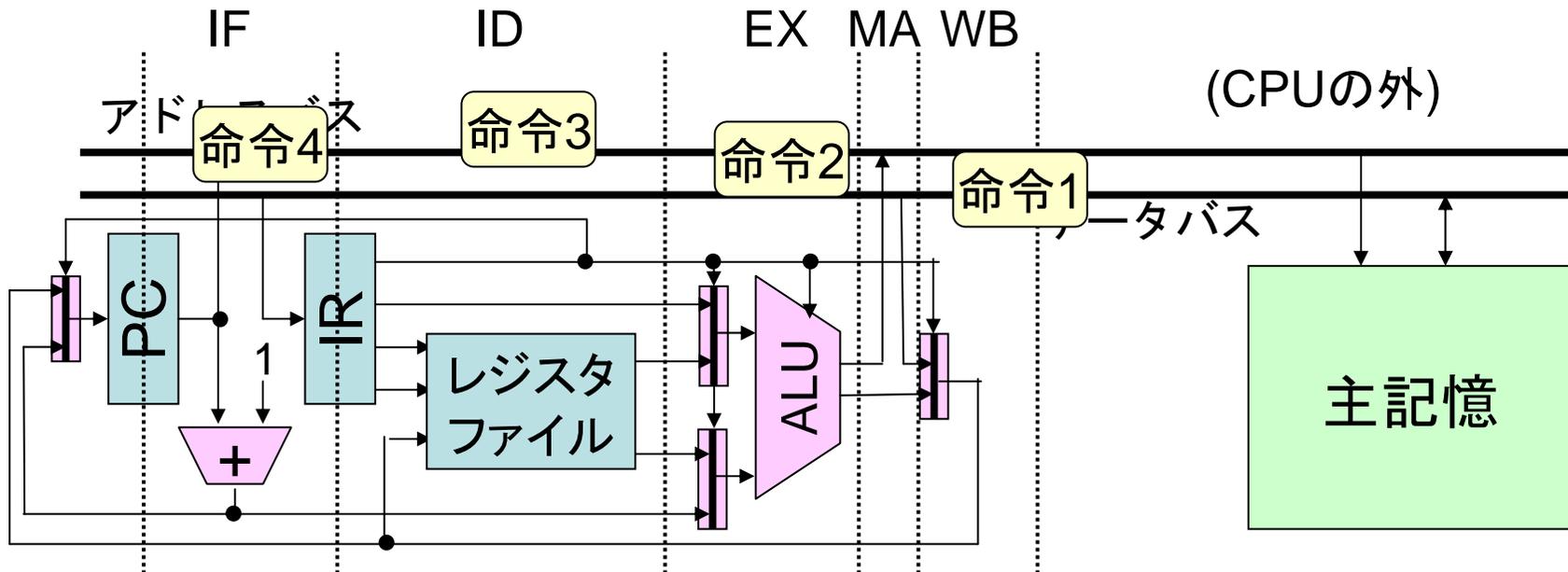
- パイプライン化ありは9クロックサイクル(CC)で終了
- パイプライン化なしは $5 \times 5 = 25$ CCで終了



データハザード

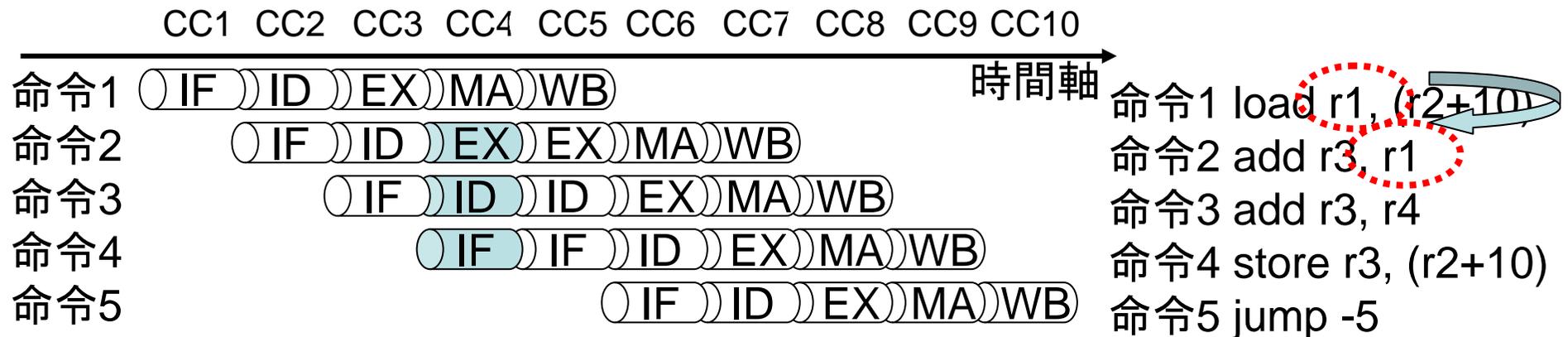
- 処理すべきデータが届いていないことが原因
 - 命令2が使うデータは命令1が主記憶読み出し中
 - 命令1のMA終了まで命令2はEXで待機
 - 後続命令も玉突きで待機

命令1 load r1, (r2+10)
命令2 add r3, r1
命令3 add r3, r4
命令4 store r3, (r2+10)
命令5 jump -5



データハザード

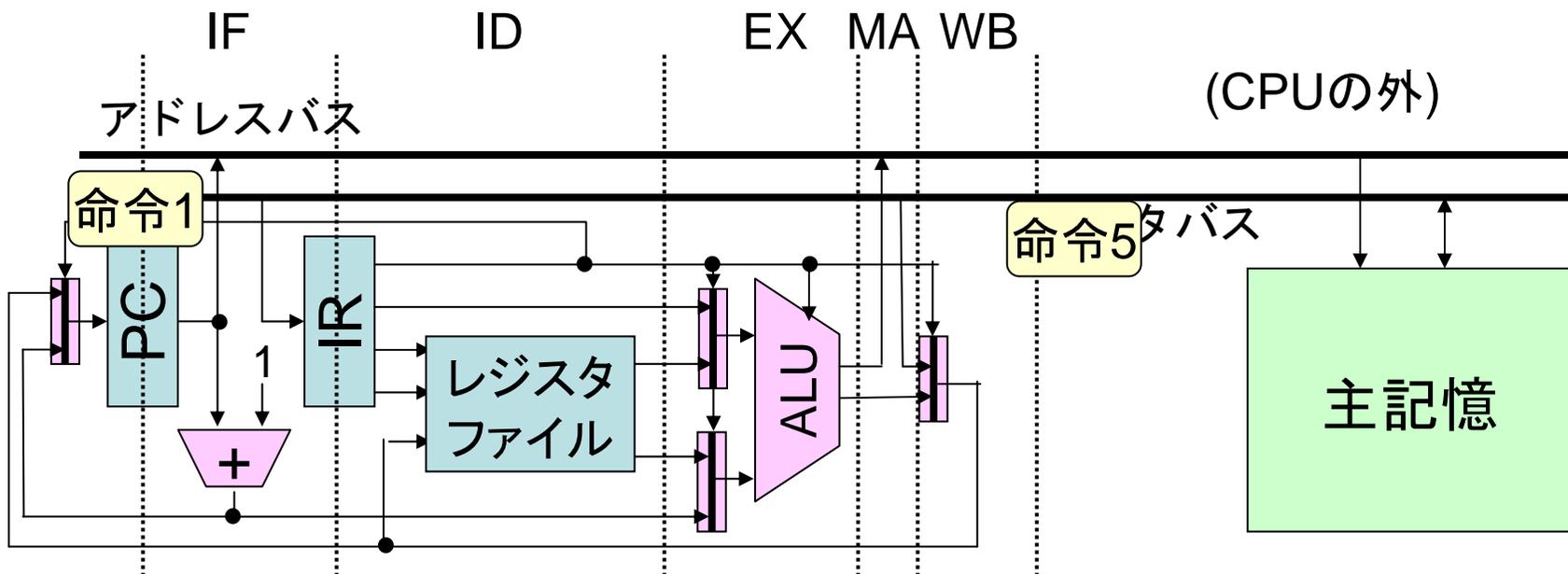
- 時間軸表記は以下のようなになる
 - 色塗り部がストール状態
 - 次CCでも同一のステージに存在
 - 終了が1CC伸びる
- **データフォワーディング**というデータハザード軽減のテクニックはすでに使われている
 - 演算結果をレジスタファイルを経さずに後続に渡す



制御ハザード

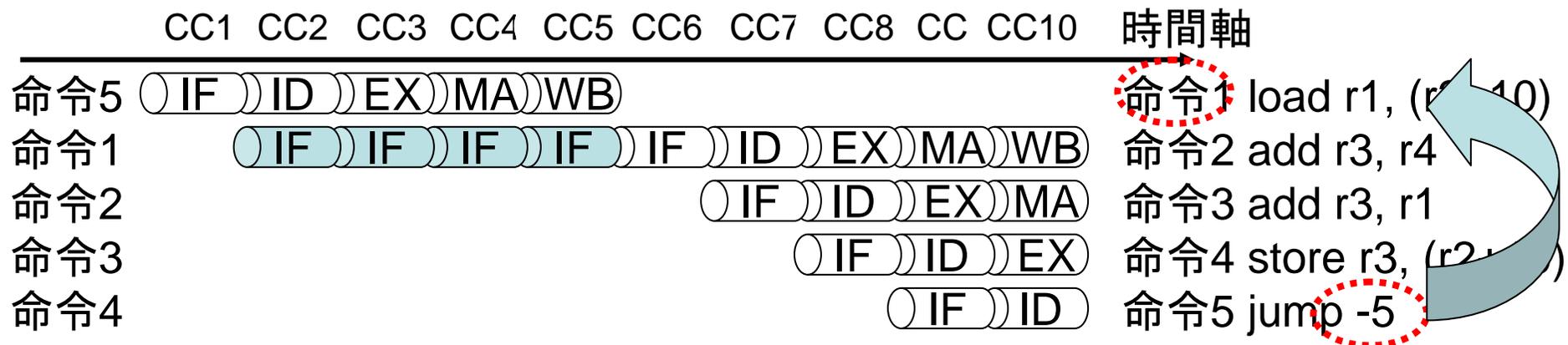
- 分岐命令の後の命令は分岐命令が完了するまでフェッチできない
 - PCが正しい値に更新されていないため

命令1 load r1, (r2 + 10)
命令2 add r3, r4
命令3 add r3, r1
命令4 store r3, (r2 + 10)
命令5 jump -5



制御ハザード

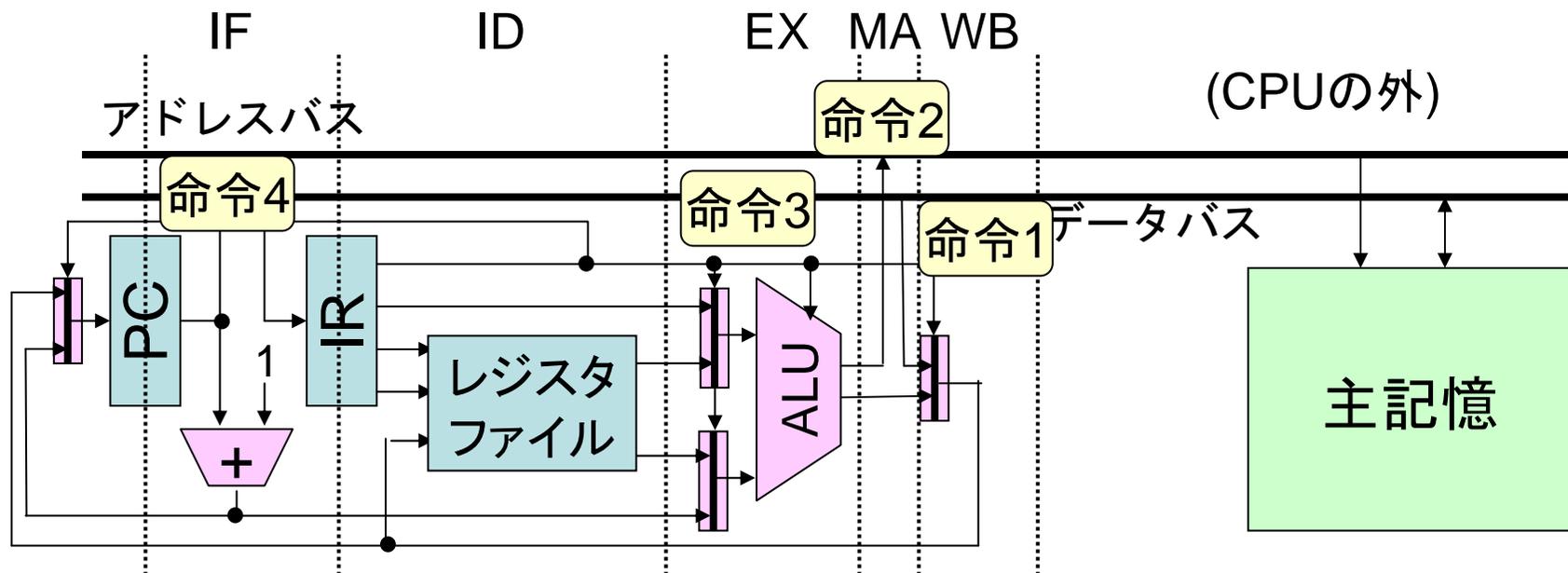
- PCが正しい値になるまで次の命令はストールする
- 分岐予測などの軽減策がある
 - 分岐先を予測してあらかじめフェッチしておく
 - パイプラインが長い(細かく分割)するほど影響大



構造ハザード

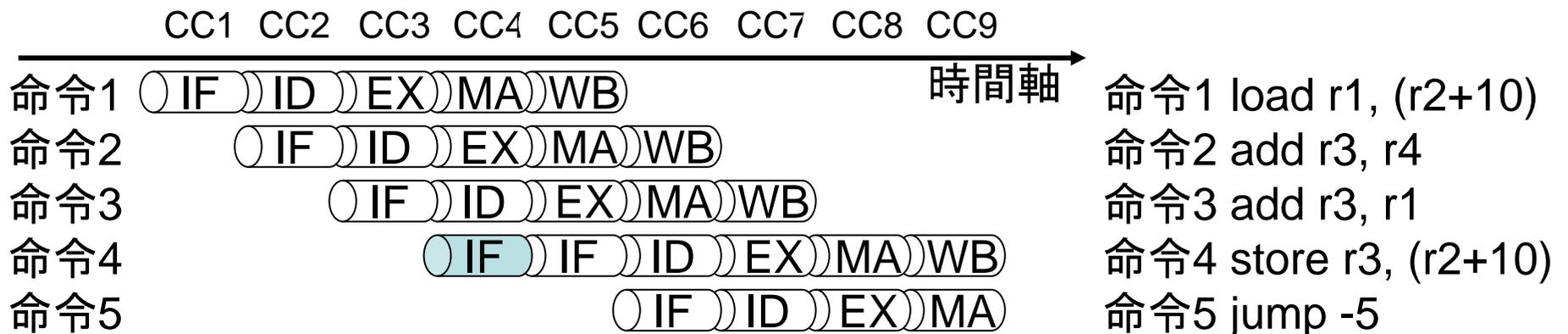
- ハードウェア数が不足することなどによるハザード
 - 下記では主記憶アクセスの競合

命令1 `load r1, (r2+10)`
命令2 `add r3, r4`
命令3 `add r3, r1`
命令4 `store r3, (r2+10)`
命令5 `jump -5`



構造ハザード

- 先の命令である命令4を優先して命令1のフェッチをストールさせる
- 対策
 - ハードウェアを増やす
 - 命令専用メモリとデータ専用メモリを準備



(b) RISCとCISC

- RISC(Reduced Instruction Set Computer)
 - 1つの機械語命令を1クロックサイクルで実行できるように揃える
 - パイプライン処理を効果的に行う
- CISC(Complex Instruction Set Computer)
 - 1つの機械語命令で複雑な動作を行う命令セット
 - パイプライン処理には向いていない
 - 命令によってパイプラインの段数が大きく変わったりする
 - 現在はRISCに近い命令に分解して実行することも多い

○ IF) ID) EX) MA) WB

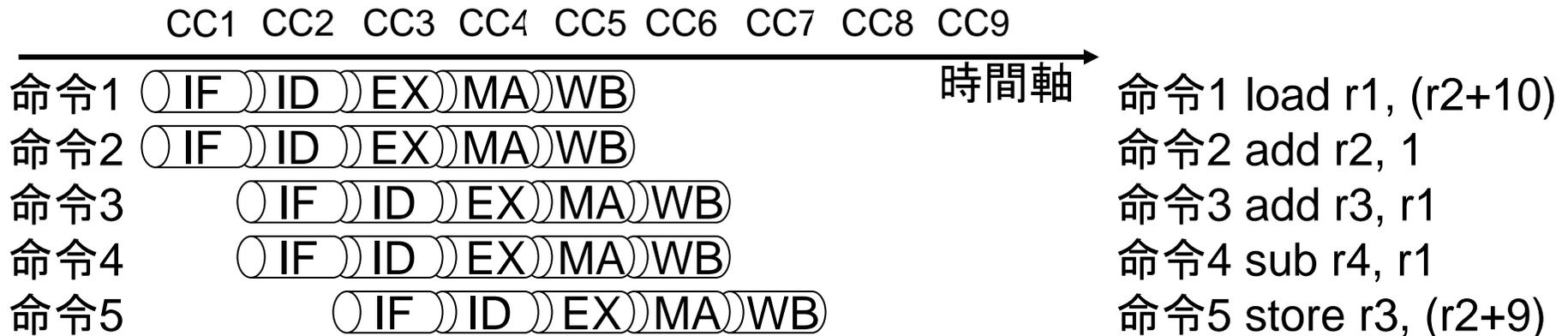
○ IF) ID) MA) MA) EX) MA) WB

命令1 add r3, r4

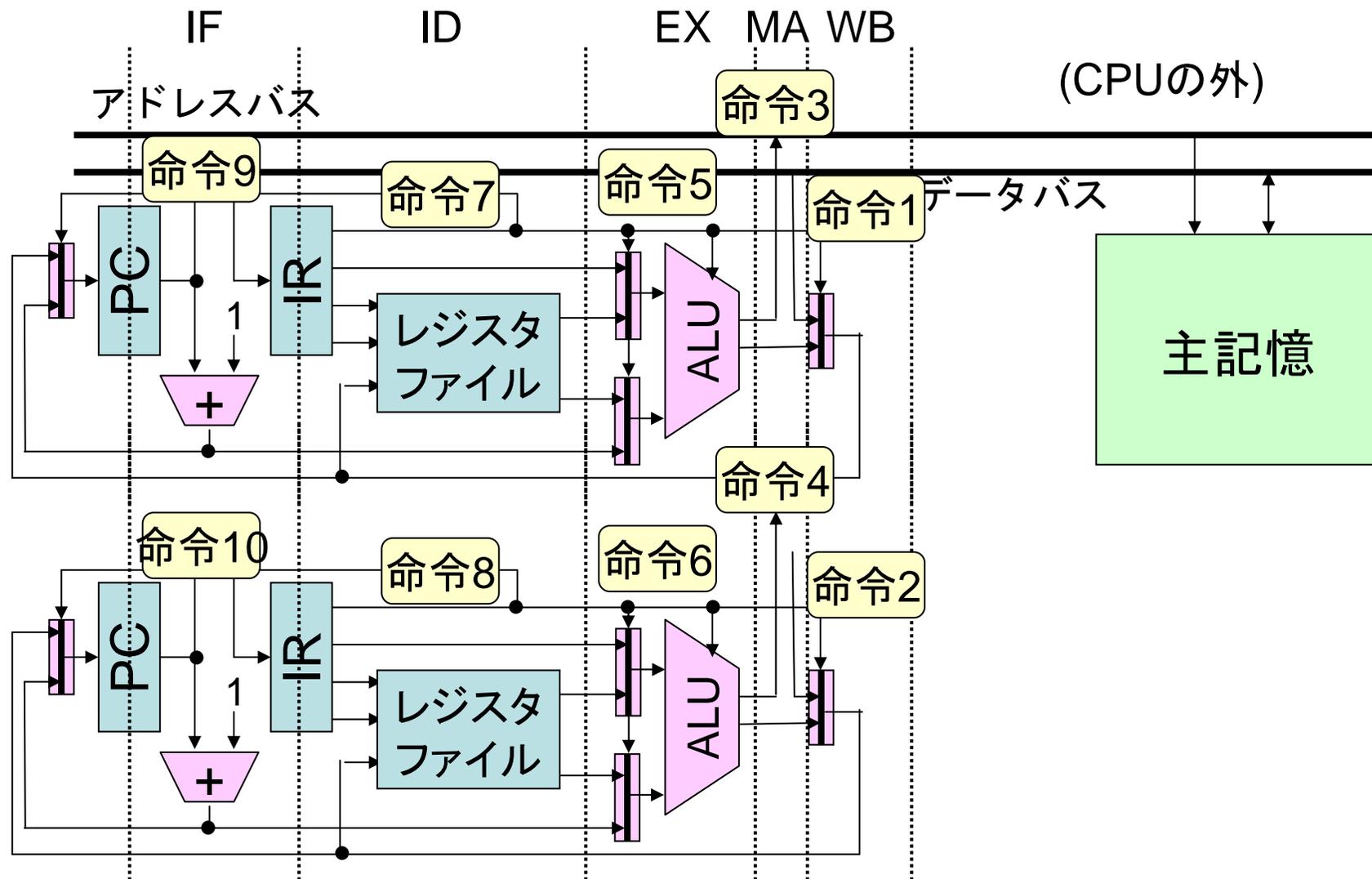
命令2 add (r3), (r4)

(c) スーパースカラ

- 命令実行に必要な論理回路を複数準備して並列実行
 - n命令並列: n-wayスーパースカラ
 - 理想的な状態では、CPIは1/nになる
- 同時に実行する演算の間でハザードが発生しないように注意する必要がある
- 3-way以上では**アウトオブオーダー実行**も併用する必要がある
 - アウトオブオーダー実行: 数十命令程度をいちどバッファに溜め、プログラム順によらず、その中から実行可能な物を実行してゆく方式



(c) スーパースカラ



(d) VLIW(Very Long Instruction Word)

- 複数の演算を定義できる命令を準備
 - 命令長が長くなる ->VLIW(命令)と呼ぶ
- VLIW命令への複数の演算の割付はコンパイラが実行
 - ハードウェアが並列実行可能か判定するスーパスカラとは対照的
 - ハードウェアが簡単になる

通常の命令

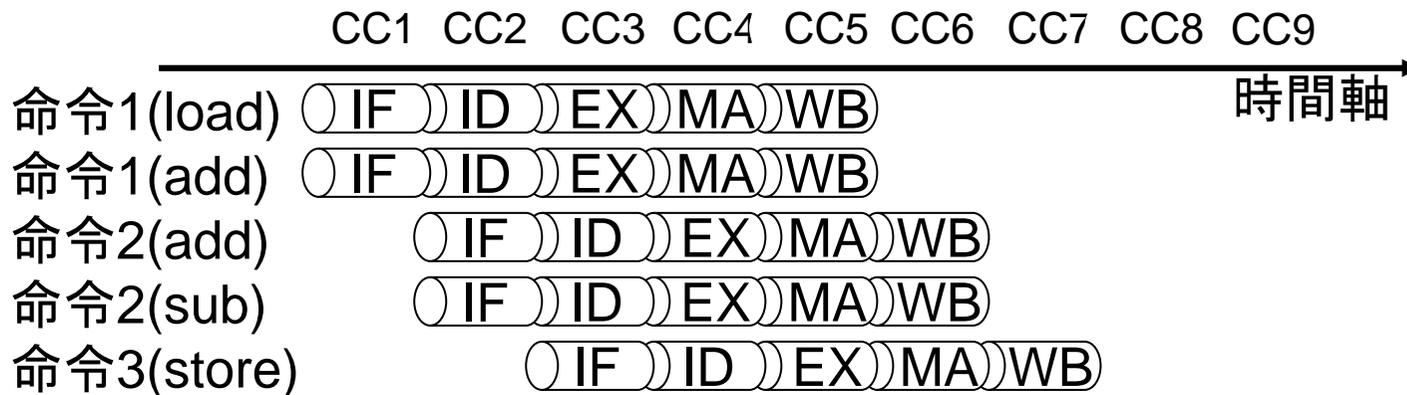
命令1 load r1, (r2+10)
命令2 add r2, 1
命令3 add r3, r1
命令4 sub r4, r1
命令5 store r3, (r2+9)

VLIW命令

命令1 [load r1, (r2+10)][add r2, 1]
命令2 [add r3, r1][sub r4, r1]
命令3 [store r3, (r2+9)][[]]

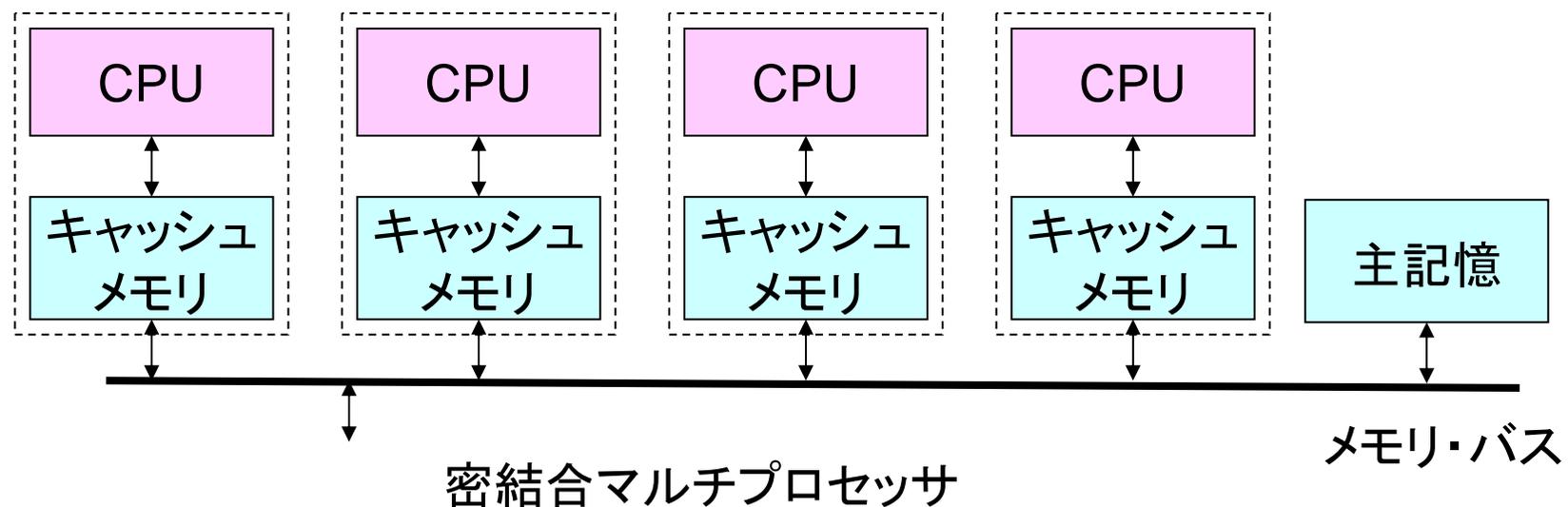
(d) VLIW(Very Long Instruction Word)

- ハードウェア構成はスーパースカラに似た形となる
- CPIは理想的には $1/n$ となる
 - n は1VLIW命令に記述できる演算の量



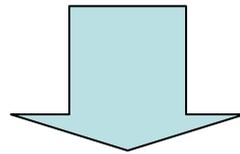
(e)マルチプロセッサ

- 複数のCPUを搭載し、処理を分散させることにより処理性能を向上させる技術
 - 疎結合マルチプロセッサ
 - 各CPUはそれぞれ主記憶装置をもち、OSも別々にできる
 - 密結合マルチプロセッサ
 - 各CPUが一つの主記憶装置を共有し、一つのOSで稼動
 - マルチコアプロセッサ



3.4 記憶階層のアーキテクチャ

- CPUが高速化しても次に処理すべきデータが来ないとうなる？
->CPUはデータを待つ(ストール)



- 記憶階層もきっちり設計しないと性能は出ない
 - 要求->読み出しまでの時間である**アクセス時間(レイテンシ)**の短縮
 - 1秒あたりのデータ転送量である**バンド幅**の拡大

参照の局所性

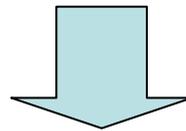
データの読み書きは分散しているわけではない

- 空間的局所性

- あるデータを読み書きした後は、近くのデータを読み書きする確率が高い

- 時間的局所性

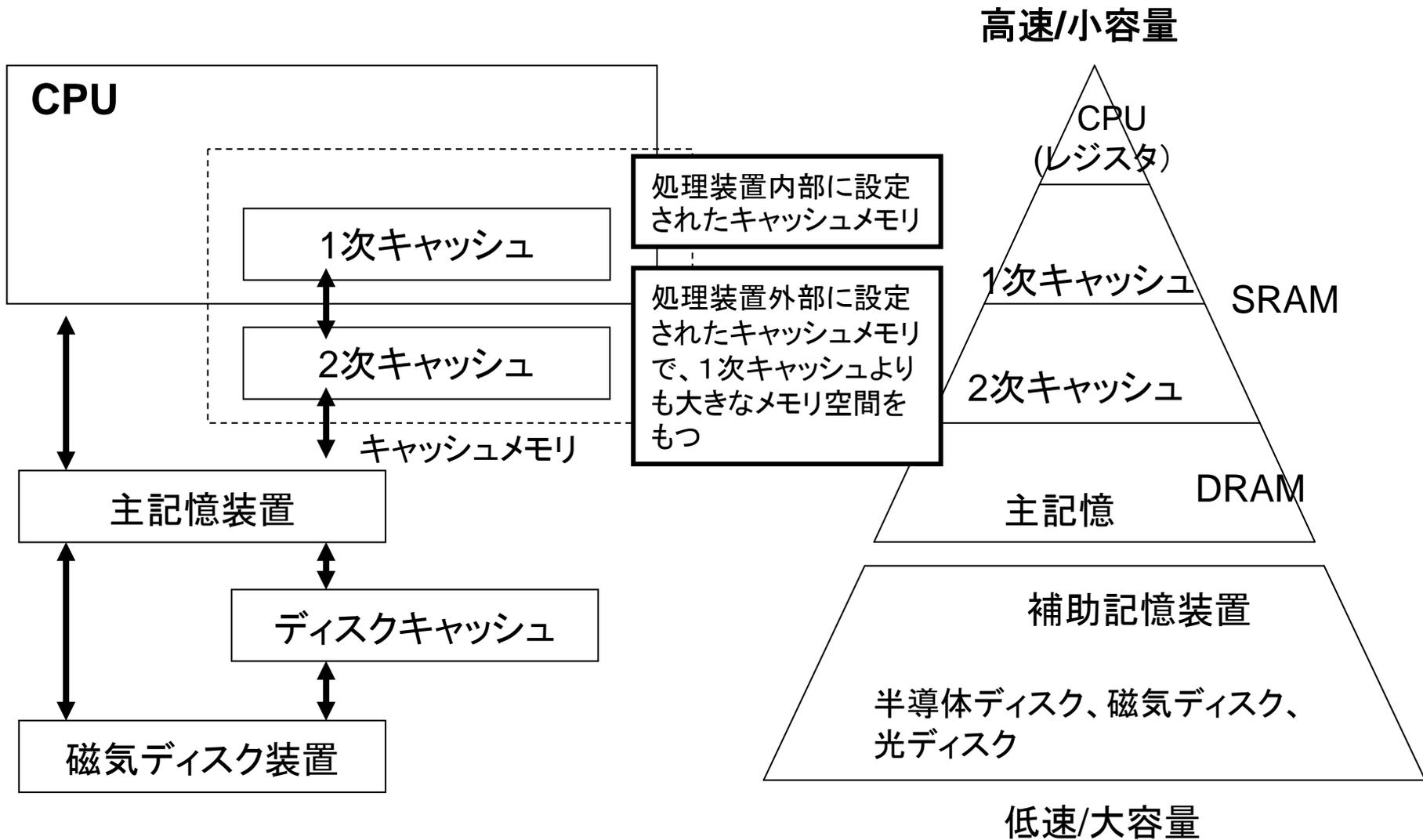
- あるデータを読み書きした後は、一定時間内に、再度そのデータを読み書きする確率が高い



- キャッシング

- 一度読み書きしたデータをまわりの物ごと高速な記憶装置に移す
- 高速な記憶装置を**キャッシュメモリ**と呼ぶ

記憶階層



最近の記憶階層の傾向

- 1次キャッシュは命令用とデータ用に分かれる
 - 狭義のハーバードアーキテクチャ
 - 本来は命令用主記憶とデータ用主記憶を作るアーキテクチャ
- 2次キャッシュもCPU内部に乗る傾向にある
- 3次キャッシュも搭載される傾向にある
 - マルチコアプロセッサでは、CPUコア間で共有
- ディスクキャッシュにSSDを使う例が出ている
 - 速度はDRAMより遅いが大容量

(2) メモリのアクセス時間

- 複雑な記憶階層では、読み出し要求の結果は記憶階層どこで返ってくるか分からない
->平均のアクセス時間を用いる

[平均アクセス時間の例1]

= 主記憶装置へのアクセス時間・(1 - キャッシュヒット率)
+ キャッシュメモリへのアクセス時間・キャッシュヒット率

例：主記憶のアクセス時間80ns、1次キャッシュのアクセス時間10ns、
1次キャッシュのヒット率90%
アクセス時間 = $80 \times (1 - 0.9) + 10 \times 0.9 = 8 + 9 = 17\text{ns}$

演習

次の記憶階層の平均アクセス時間を求めよ

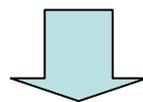
- 1次キャッシュで読み出される確率80%
- 1次キャッシュは1nsでアクセス可能
- 2次キャッシュで読み出される確率10%
- 2次キャッシュは10nsでアクセス可能
- 主記憶で読み出される確率10%
- 主記憶は100nsでアクセス可能

平均アクセス時間は、各階層にアクセスされる確率と各階層のアクセス時間の積の総和となる。

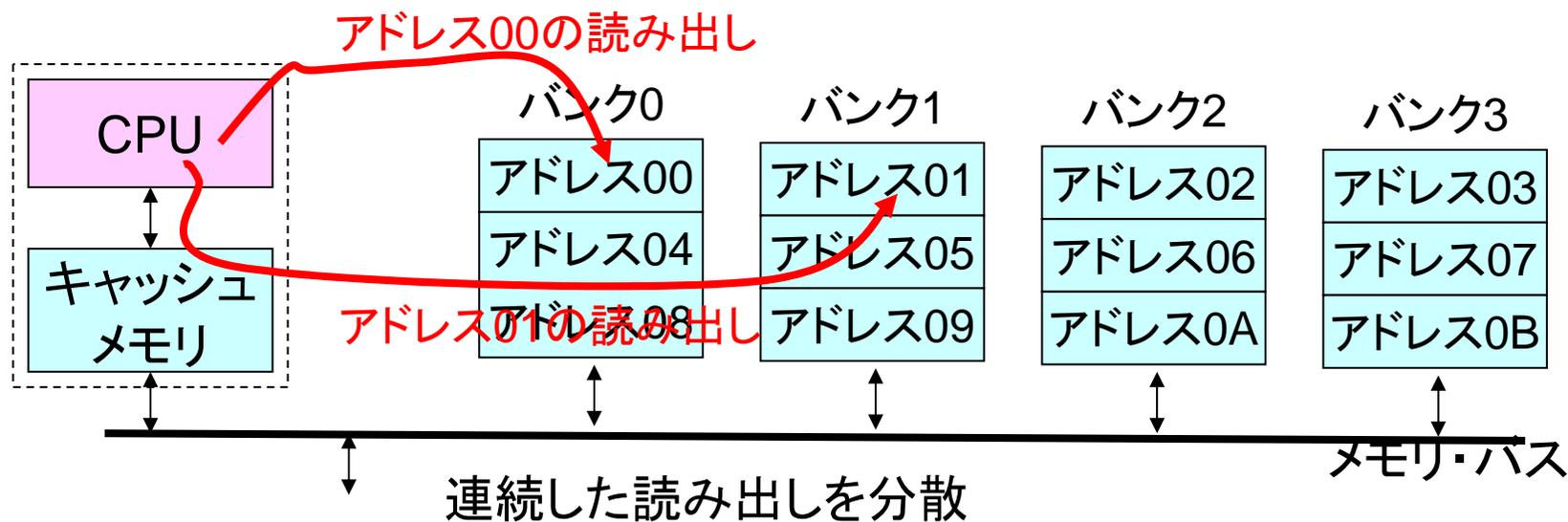
$$\begin{aligned}\text{平均アクセス時間} &= 0.8 \times 1 + 10 \times 0.1 + 100 \times 0.1 \\ &= 0.8 + 1 + 10 = 11.8\text{ns}\end{aligned}$$

(3) メモリインタリーブ

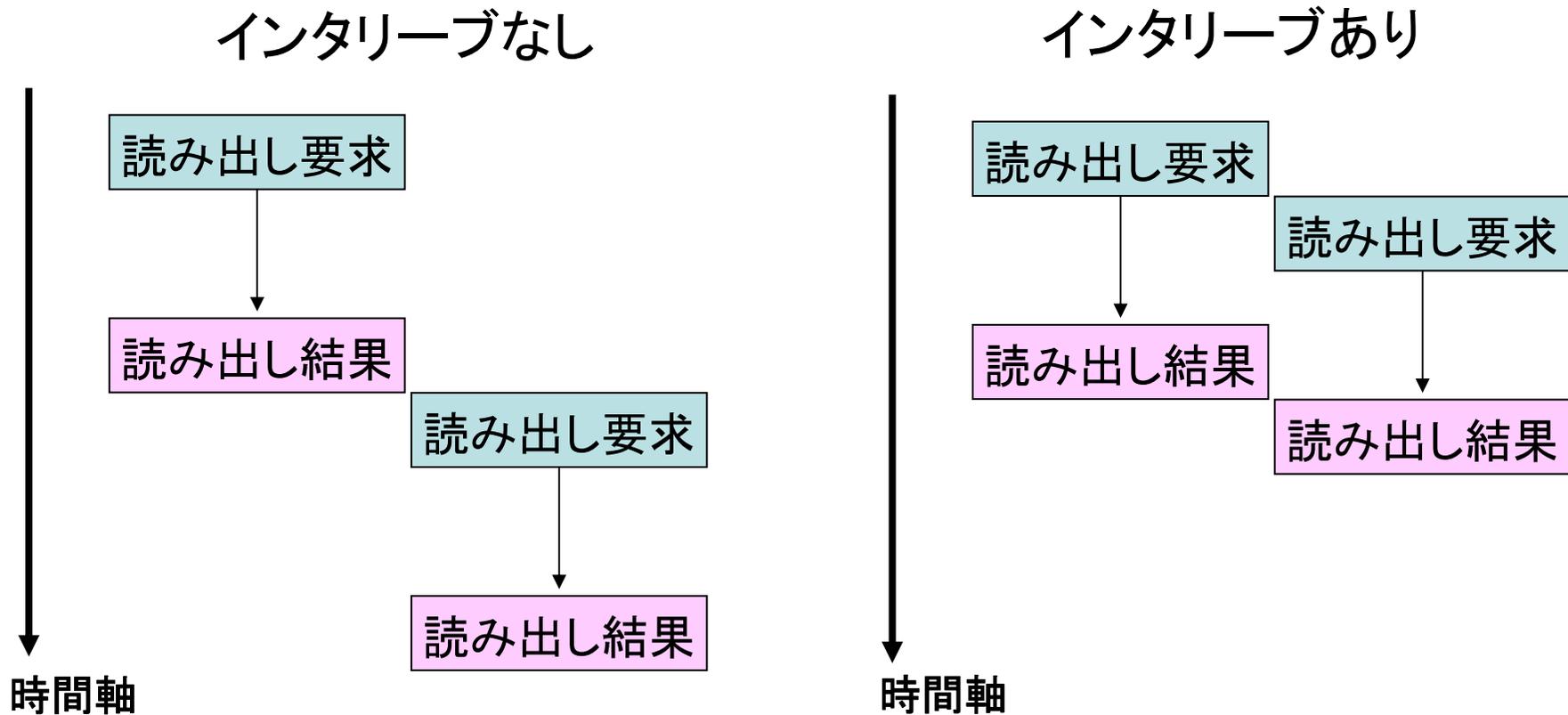
- CPUよりも主記憶の方が動作速度が遅い
 - 連続して読み出しをかけると待たされることが多い
 - 1次キャッシュあたりまでは連続読み出し可能



- 主記憶を**バンク**に分割して並列に読み書き可能にする
 - さらに連続した読み出しを分散させる



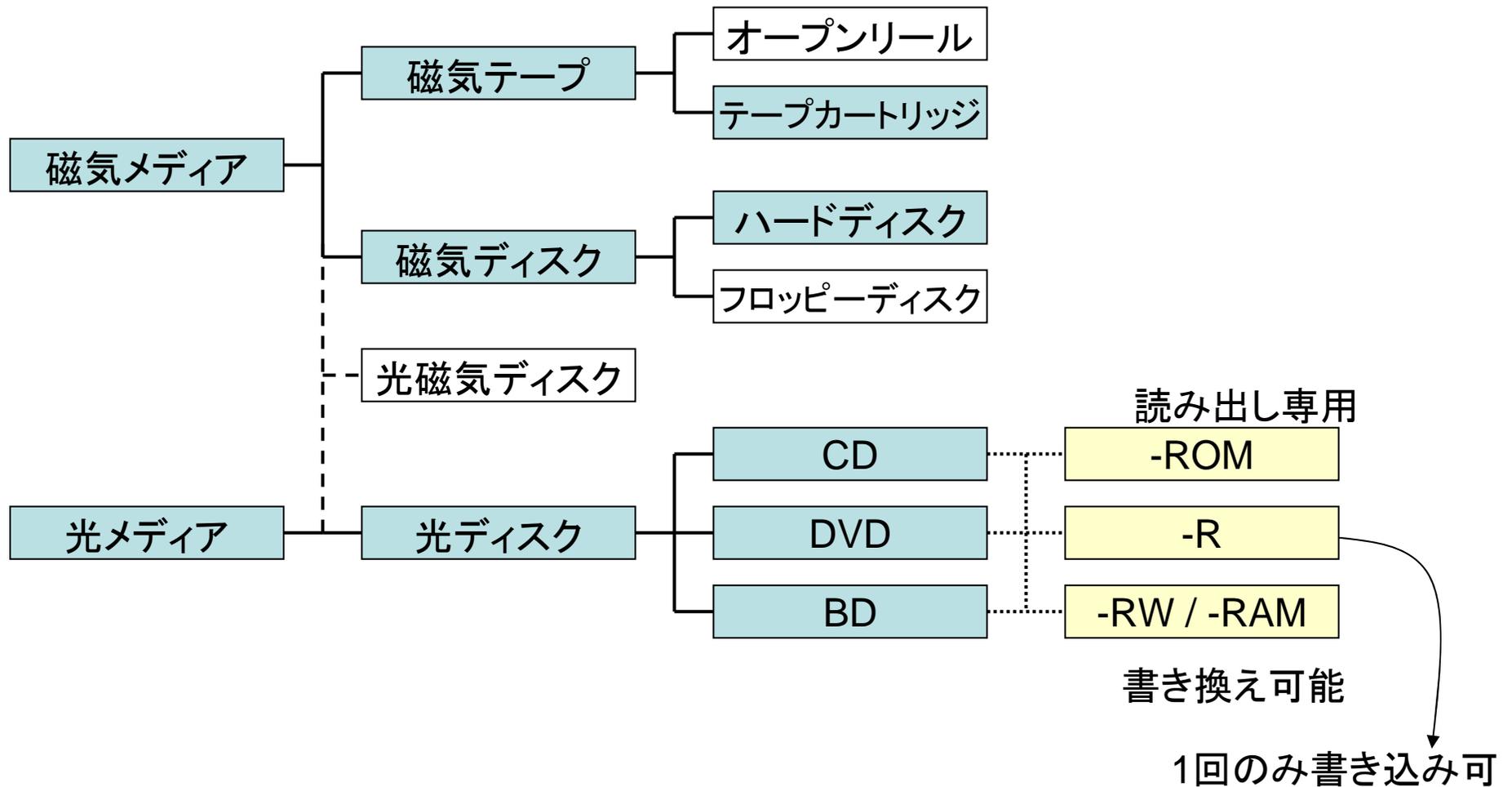
(3) メモリインタリーブ



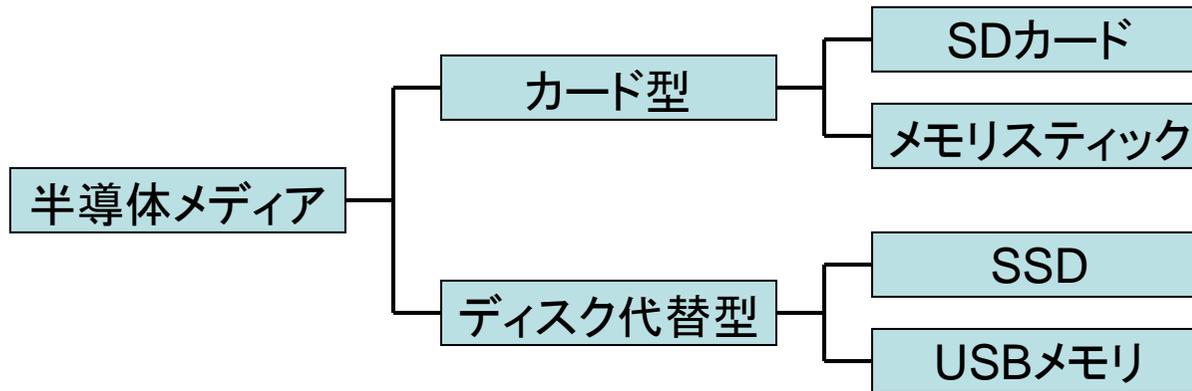
- あくまで理想状態の結果で、アドレス00->04->08のようなアクセスには効果がないことに注意

3.5. 入出力装置としての補助記憶装置

様々な補助記憶装置



様々な補助記憶装置



- 新たなメディアによって廃れる記憶装置もある
 - そもそも、メジャーになれなかったものも多数
- 近年では半導体メディアが勢力拡大中

情報の記録方法の違い

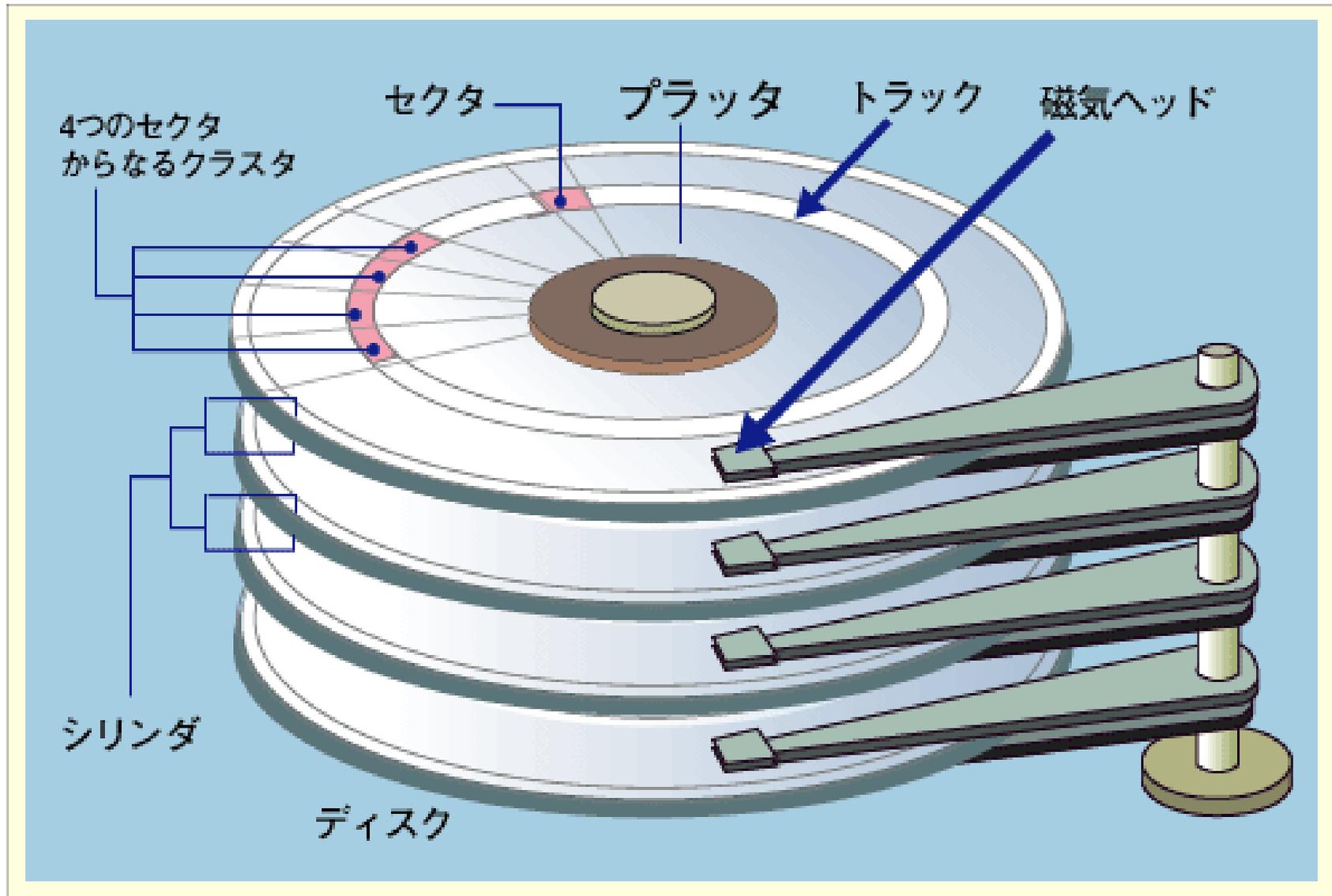
- 磁気メディア：磁気の方角(NS/SN)で0/1を記録
- 光ディスク：メディアにレーザー光を照射して、その反射率の違いを0/1とする
- 光磁気ディスク：メディアにレーザー光を照射して、その偏光の違いを0/1とする
- 半導体メディア：フラッシュメモリなどの不揮発性のICメモリに記録

(1)磁気メディア

(1.1) 磁気ディスク(ハードディスク、HDD)

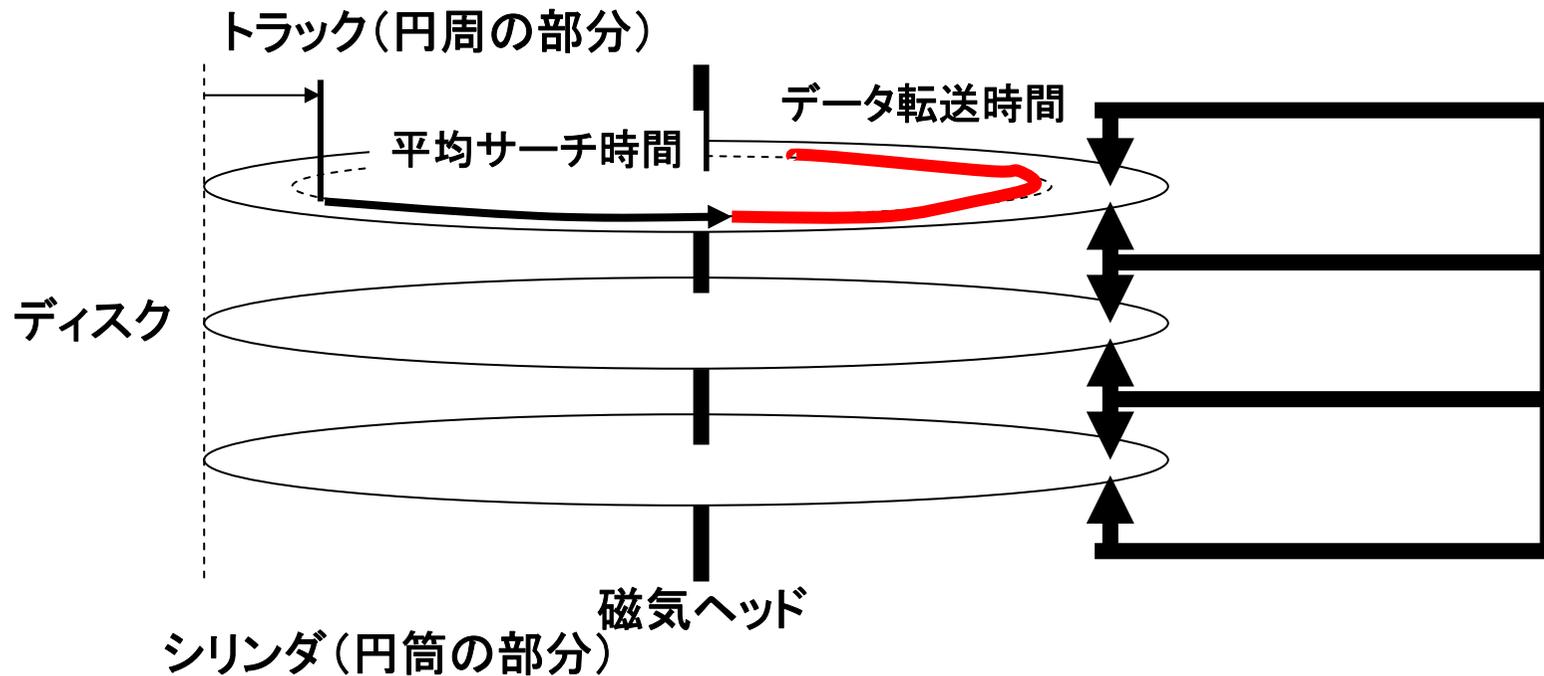
- データ構造
 - ディスクの同心円状にいくつかのトラックを配置
 - トラックを扇状にいくつかのセクタに分割
 - 複数のディスクを配置し、同一の位置のトラックの集合をシリンダとする
 - 磁気ヘッドはディスク間で同期して動く
- データはセクタ単位で書かれる
 - いくら小さなデータでも最低1セクタは占有
- 最近は必ずしも上記の構造になってはいない
 - 内周と外周でトラックあたりのセクタ数が違ったりする
 - 各セクタにシーケンシャルな番号をつけて管理
 - 内部的には上記のような処理もしている

磁気ディスクの構造



磁気ディスクの平均アクセス時間

[ブロックのアクセス時間]
= [平均シーク時間] + [平均サーチ時間] + [データ転送時間]

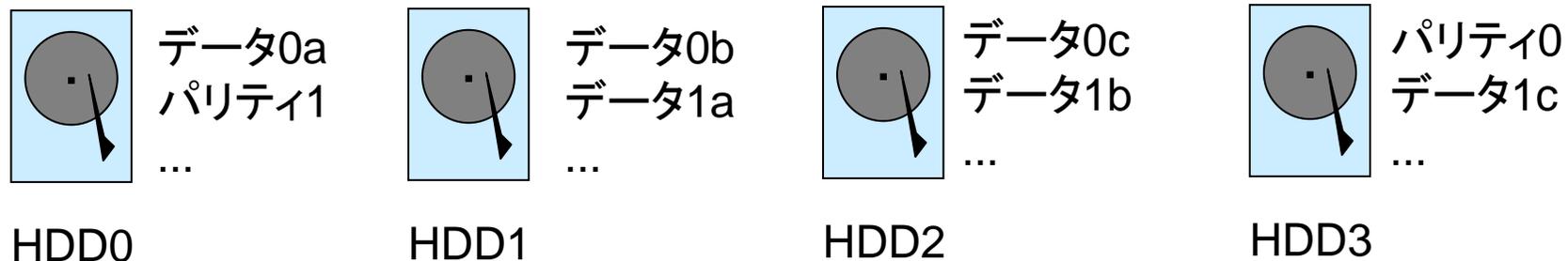


磁気ディスクの平均アクセス時間

- 平均シーク時間
 - 磁気ヘッドがデータブロックのあるトラックまでの平均移動時間
- 平均サーチ時間
 - 磁気ヘッドがデータブロックの先頭セクタに来るまでの平均時間
 - 最大のサーチ時間は1回転の時間であり、最小のサーチ時間は0
[平均サーチ時間] = [1/2回転の時間]
- データ転送時間
 - 磁気ヘッドがデータブロックを読み込む時間
 - 通常、ディスク回転速度とブロック長(セクタ数)から計算
 - データブロックが別トラックにまたがれば、再度、シークやサーチが必要

(2) RAID (Redundant Array of Independent (Inexpensive) Disks

- 複数のHDDを集めて一つの装置として使う方法
 - 冗長化: どれか1つのHDDが壊れてもデータは無事
 - RAID1(ミラーリング): 同一のデータを複数台のHDDに記録
 - 高速化: データを分散させて高速読み出し
 - RAID0(ストライピング): メモリインタリーブと同じ
- 両方を目的としたRAID5
 - **パリティ**という誤り訂正符号を生成
 - データとパリティを分散して書き込む



さらに高規格なRAIDと RAIDの注意点

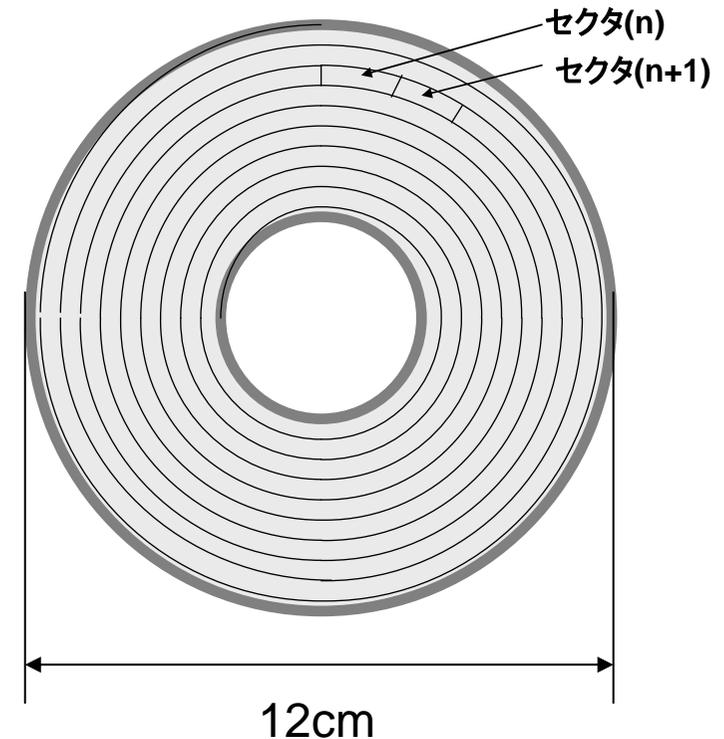
- 高規格なRAID
 - RAID6
 - パリティを2重に取って2個のHDD故障に対応
 - RAID10
 - RAID1とRAID0の併用
 - 同一のバンクに対応するHDD2台が同時に壊れない限りOK
- RAIDの注意点
 - 動作継続のためであって、バックアップのためではない
 - 誤って削除とかOSが暴走してデータ上書きなどは防げない
 - 同一生産ロットのHDDで構成するとHDDが同時に壊れることがある

(3) 光ディスク

- データ構造はHDDを片面にした感じ
- ただし、読み出し専用を基本にしている
 - 書き込みはあまり得意でない
 - 基本的に追記しかできない
 - 追記スペースがなくなったら全部消して書き直し
- 系統を立てたモデル展開をしている
 - 容量の違い: CD, DVD, BD
 - 読み書き仕様: -ROM, -R, -RW, (-RAM)

容量の違いによる光ディスク規格

- CD: Compact Disc
 - 約650MBもしくは約700MB
 - 音楽のデジタル記録を目的に開発
- DVD: Digital Versatile Disc
 - 1層約4.7GB、2層約8.5GB
 - 映像のデジタル記録を目的に開発
- BD: Blu-ray Disc
 - 1層約25GB、2層約50GB
 - High Definition映像のデジタル記録を目的に開発



読み書き仕様の違いによる 光ディスク規格

- (CD|DVD|BD)-ROM: Read Only Memory
 - 樹脂製の円盤に細かい凹凸をつけてデータを記録
 - 凹凸による反射光の違いをデータとして読み取る
- (CD|DVD|BD)-R: Recordable
 - 記録面に添付されている有機色素膜にレーザー照射で穴を開け、データを記録
 - 有機色素膜の下には反射膜があり、反射光に差異をつける
- (CD|DVD|BD)-RW: ReWriteable
 - データ記録膜の結晶状態／アモルファス状態の違いで0/1を表す
 - 強いレーザー光で急速加熱／急冷でアモルファス化、弱いレーザー光で焼きなまして結晶化

光ディスクの書き込み

- ディスクアットワンス
 - ディスク単位で書き込みを行う形式
 - 一度だけしか書き込みできず、追記は不可能
- マルチセッション
 - 複数のセッション(CDの書き込み単位)の書き込みができる追記が可能な形式
 - トラックアットワンス(トラック単位での書き込み)で記録される
- パケットライティング
 - トラックよりも小さいパケット単位で書き込みが行える方式

その他の補助記憶装置

- フロッピーディスク(フレキシブルディスク)
 - ディスク1枚だけのHDDと考える(容量は1.44MB)
- 光磁気ディスク(Magneto Opticalディスク)
 - 強いレーザーを照射しつつ磁界をかけることでデータを磁気的に書き込む
 - 読み出しは磁化の向きで弱いレーザーが偏光されることを利用
- 磁気テープ
 - トラックが1次元のHDDと考える
 - データバックアップ用として利用
- 半導体メディア
 - ICメモリに同じ
 - OS側からはディスクと見えるようにディスクを模している

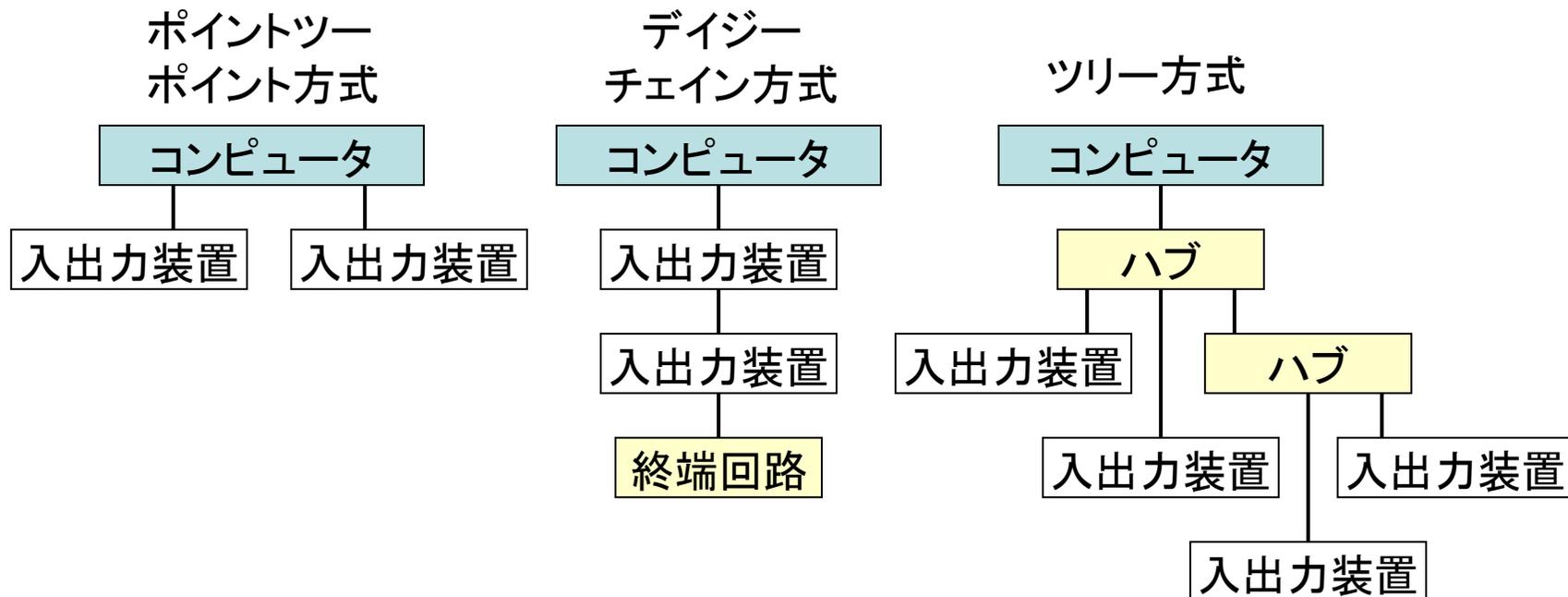
4.3 入出力インターフェース

入出力インターフェース: コンピュータと入出力装置や補助記憶装置の接続規格

- シリアルインタフェース: 単一の伝送路を利用
- パラレルインタフェース: 複数の伝送路を利用
- ワイヤレスインタフェース: 伝送路なし・・・のように見えるが、実質は単一の伝送路
 - 電波の飛ぶ空間は共有されるため

入出力インターフェースによる コンピュータと装置の接続形式

- ポイントツーポイント方式：S-ATA, SASなど
- デイジーチェーン方式：SCSI, GPIBなど
- ツリー方式：USB, IEEE1394など



(1) シリアルインターフェース

- 特徴
 - 単一の伝送路を利用して1ビットずつデータを伝送
 - 近年では超高速化に有利なため、様々なインターフェースがシリアル化
- 代表的なもの
 - RS-232C
 - 様々な周辺機器を接続するための低速な汎用インターフェース
 - USBにその役割を譲り渡した
 - 実装がシンプルなので、組み込み機器などでは未だに使われている
 - USB(Universal Serial Bus)
 - 周辺機器をパソコンと接続する高速なシリアルインターフェース
 - 高速化で記憶装置も接続する傾向にある
 - ハブを介してツリー状に接続できる
 - ホットプラグ機能(機器を動作中に抜き差しできる)を有する

(1) シリアルインターフェース

- 代表的なもの(続き)
 - IEEE1394
 - USBと異なり、パソコンを介さず、機器同士で接続可能
 - デジタルビデオカメラの接続端子として採用(i.LINK、DV端子)
 - シリアルATA (S-ATA)
 - パラレルインターフェースで説明するATAのシリアル版
 - 超高速化のためにシリアル化した物の1例
 - SAS(Serial Attached SCSI)
 - パラレルインターフェースで説明するSCSIのシリアル版

(2) パラレルインタフェース

- 特徴
 - 複数の伝送路を束ね、同時に複数ビットの伝送
 - かつては高速伝送の主流であった
 - 超高速では複数の伝送路間の同期が難しい ->シリアル化
- 代表的なもの
 - ATA(Advanced Technology Attachment), IDE
 - もともと内蔵用のHDDを接続するためのパラレルインタフェース(IDE)
 - CD-ROM等を含めて4台まで接続できるように拡張(ATA)
 - SCSI(Small Computer System Interface)
 - デイジーチェーン接続で最大7台まで周辺装置を接続
 - ワークステーションあたりが主に用いていた
 - GPIB(IEEE488)
 - 計測機器などのデジタル機器を接続する

(3) ワイヤレスインタフェース

- 特徴

- 配線を行う必要がない
- 通信が混雑していても何が悪いのか分かりにくい

- 代表的なもの

- IrDA

- 赤外線を利用した近距離のデータ通信インタフェース
- 送信部と受信部を向かい合わせにしないと通信できない

- Bluetooth

- 携帯機器向けの電波を用いた無線インタフェース
- 免許なしで自由に使える2.4GHz帯の電波を利用
- 到達距離は10m程度

3.7 入出力装置

(1) 入力装置

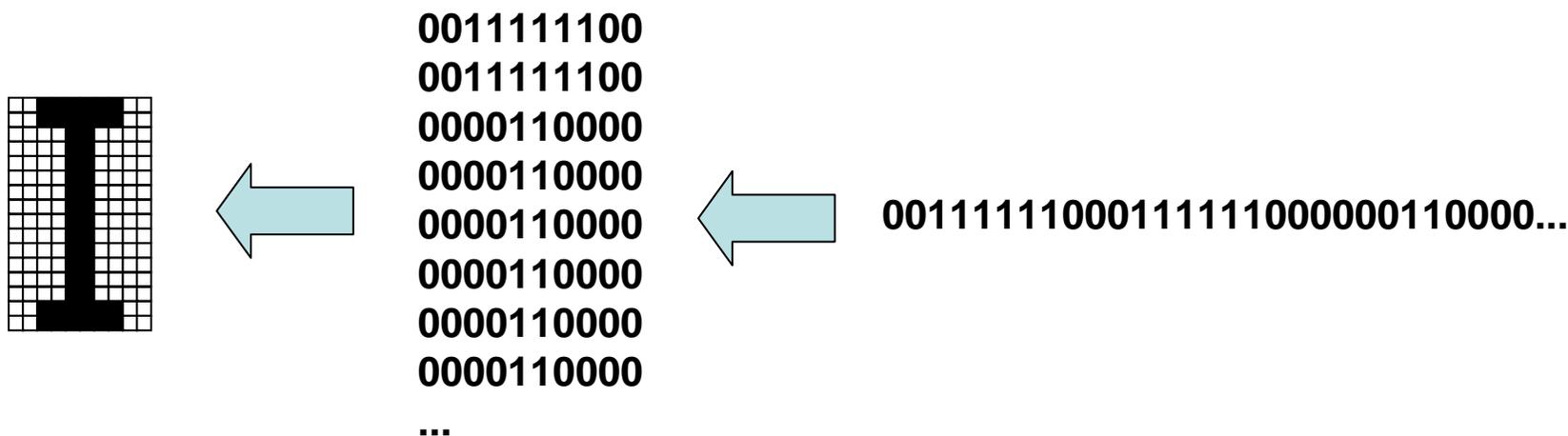
- キーボード
 - キーをタイプすることにより文字、数字、記号を入力
 - 最近はUSBで接続
- ポインティングデバイス
 - ディスプレイ画面上に位置情報(座標情報)を入力する装置の総称
 - 最近はUSBで接続
 - マウス、トラックボール、ペンタブレット、タッチパネル、など
- イメージスキャナ
 - 図形や写真を読み取って、画像データに変換して入力する装置
 - 性能指標として、1インチ当たりのドットの数であるDPI(Dots Per Inch)を用いる(高いほど高性能)

(1) 入力装置

- OCR(Optical Character Reader)
 - 手書きの文字や印刷された文字を光学的に読み取り、文字データに変換する
 - 誤認識に注意
- OMR(Optical Mark Reader)
 - マークシート用紙などに、マークされた記入欄を光学的に読み取る装置
 - OCRよりもはるかに誤認識が少ない
 - バーコードを読み取る装置であるバーコードリーダーも、OMRの1種

(2) ディスプレイ

- メモリ(VRAM)中のデータを画像として表示
 - VRAM(Video Random Access Memory)と呼ばれる
 - 1画面のデータ量は、VRAM容量以下
- 画面のサイズ(画素量)は規格名を用いることも
 - 例: VGAは640x480, XGAは1024x768



画像(画面)のデータ量の計算

- [画像のデータ量] = [画像の総画素数] × [1ドットあたりの容量]
- [画像の総画素数] = [縦の画素数] × [横の画素数]
- [1ドットあたりの容量] = [赤色情報のビット数]
+ [緑色情報のビット数] + [青色情報のビット数]
+ ([輝度情報のビット数])

例:

[100ドット × 500ドットで16777216色の画像]
=[総画素数50,000ドット] × [RGB各8ビット]
=[50,000ドット] × [24ビット]
=1,200,000ビット
=150,000バイト

演習

以下の画面の表示に必要な最低限のVRAMの量は？

- 画面サイズはXGA
- 色はRed 5ビット、Green 6ビット、Blue 5ビットで表示

XGAの画面サイズは1024 x 768なので、総画素数は、

$$\text{総画素数} = 1024 \times 768 = 786432$$

となる。

1画素あたりのビット数は16ビットなので、必要最低限のVRAMの量は、

$$\text{VRAM量} = 786432 \times 16 = 12,582,912 \text{ビット} = 1,572,864 \text{バイト}$$

ディスプレイの種類

- CRTディスプレイ (CRT: Cathode Ray Tube)
 - ブラウン管を利用したディスプレイ
 - かつては主流であったが、現在は廃れている
- 液晶ディスプレイ(LCD: Liquid Crystal Display)
 - 電圧をかけると分子の配列が変わる液晶の特徴を利用
 - 2枚のガラス板の間に液晶を封入し、電圧をかけることで、液晶分子の向きを変え、光を透過させたり反射させたりすることで画像を表示
 - 自分では発光しないので、バックライトを配置したりする

ディスプレイの種類

- 液晶ディスプレイの種類
 - STN型: 単純マトリックス方式
 - 低コストだが、TFT型に比べて応答速度が遅く、コントラストも低い
 - TFT型: アクティブマトリックス方式
 - 画素ごとにトランジスタなどの「アクティブ素子」を配置
 - 現在のコンピュータのディスプレイで広く使用されている
- 有機ELディスプレイ(EL: Electric Luminescence)
 - 電圧をかけると発光するEL素子を配置し、それに電圧をかけることで発光
 - 電圧をかけると発光するEL素子を配置し、それに電圧をかけることで発光

(3) プリンタ

- 特徴
 - 紙に文字や画像を出力する出力装置
 - 性能指標
 - 1秒あたりに印刷できる文字数(1分あたりに印刷できるページ数)
 - 1インチあたりの画素数(DPI: Dots Per Inch)などがある
- 代表的なもの
 - ドットインパクトプリンタ
 - 細かなピンを多数配置し、そのピンで文字や図形に合わせてインクを染み込ませたインクリボンを紙に叩きつける
 - カーボン紙などの複写用紙に対応できる

(3) プリンタ

- 代表的なもの(続き)
 - インクジェットプリンタ
 - インクを吐出できる細かなノズルを多数配置し、そのノズルから文字や図形に合わせてインクを吐出
 - レーザープリンタ
 - レーザー光を当てると帯電するドラムにレーザー光で文字や図形を描き、帯電部にトナーと呼ばれるインクの粒子を付着させ、それを紙に転写する (コピー機と同じ)
 - 印刷をページ単位で処理するので、性能指標もページ単位
 - プロッタ
 - 紙を画板に固定し、ペンを動かして図形を描くこと専門とする
 - 現在では、ペンの代わりにインクジェットのヘッドを用い、紙の側も動かす物が主流

3章のまとめ

- コンピュータの機能は5つに分けて考えられる
- CPU内部で命令と情報が処理される様子
 - 命令は基本的にIF, ID, EX, MA, WBの5段階で実行される
 - 主記憶中の情報の位置を効果的に示すために様々なアドレッシングモードがある
 - CPU性能はクロック周波数や平均CPIなどで見積もれる
- CPUへ効率的にデータを供給するため、記憶階層が構築されている
 - 記憶階層の性能は平均アクセス時間やバンド幅であらわされる
- 入出力関係
 - ディスク性能や画面データの量について

