

コンピュータの仕組み (4)

柴山 潔

コンピュータの仕組み

- 1 コンピュータシステム
- 2 ハードウェア
- 3 内部装置
- 4 プロセッサ(1)
- 5 メモリ
- 6 外部装置
- 7 論理回路
- 8 オペレーティングシステム

4 プロセッサ(1)

4.1 プロセッサの動作とその制御(1)

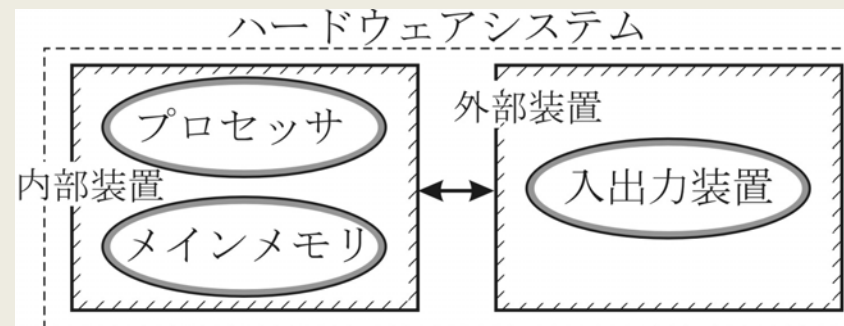
4.2 プロセッサの機能 —演算と制御—

4.3 演算器

4.4 順序制御機構

主要なハードウェア装置

- プロセッサ
 - メモリ, メインメモリ
 - 入出力装置
- これら3点のハードウェア装置がそれぞれの役割を果たしてはじめて、ハードウェア機構はシステムとして動作



(重要)

(再掲)

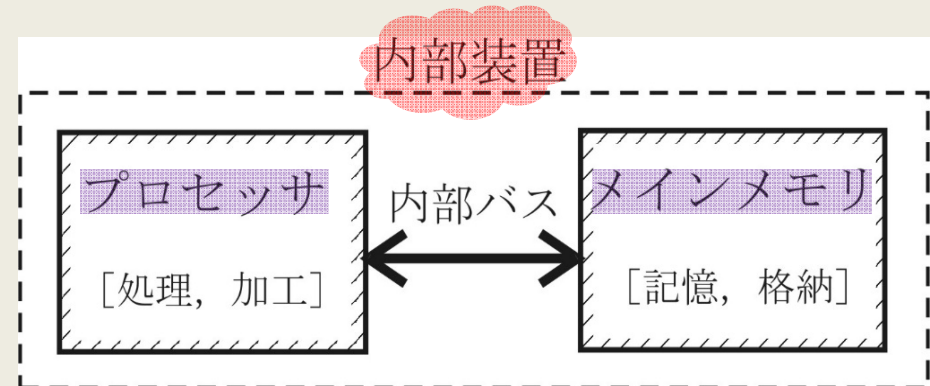
プロセッサとメインメモリ

■ 内部装置 (=コンピュータ本体)

- 簡単な計算から高度な情報処理までを実行するハードウェア機構
- 次の2種類のハードウェア装置を組み合わせたハードウェアシステム

■ **プロセッサ**: 情報を**処理**,
その処理の仕方を**制御**

■ **メインメモリ**: 情報を**記憶**,
格納



プロセッサの機能

- プロセッサでは、メインメモリから読み出してきたマシン命令にしたがって、演算器による計算や処理を行ったり、順序制御機構によって実行すべき命令を決定

マシン命令間でのデータの共用

◆ 複数マシン命令間でデータを共用する場合

(例) 命令Aが生成した演算結果XをAの次(直後)の命令Bのソースオペランドとして使う場合

- Aの結果オペランドXをメインメモリと指定

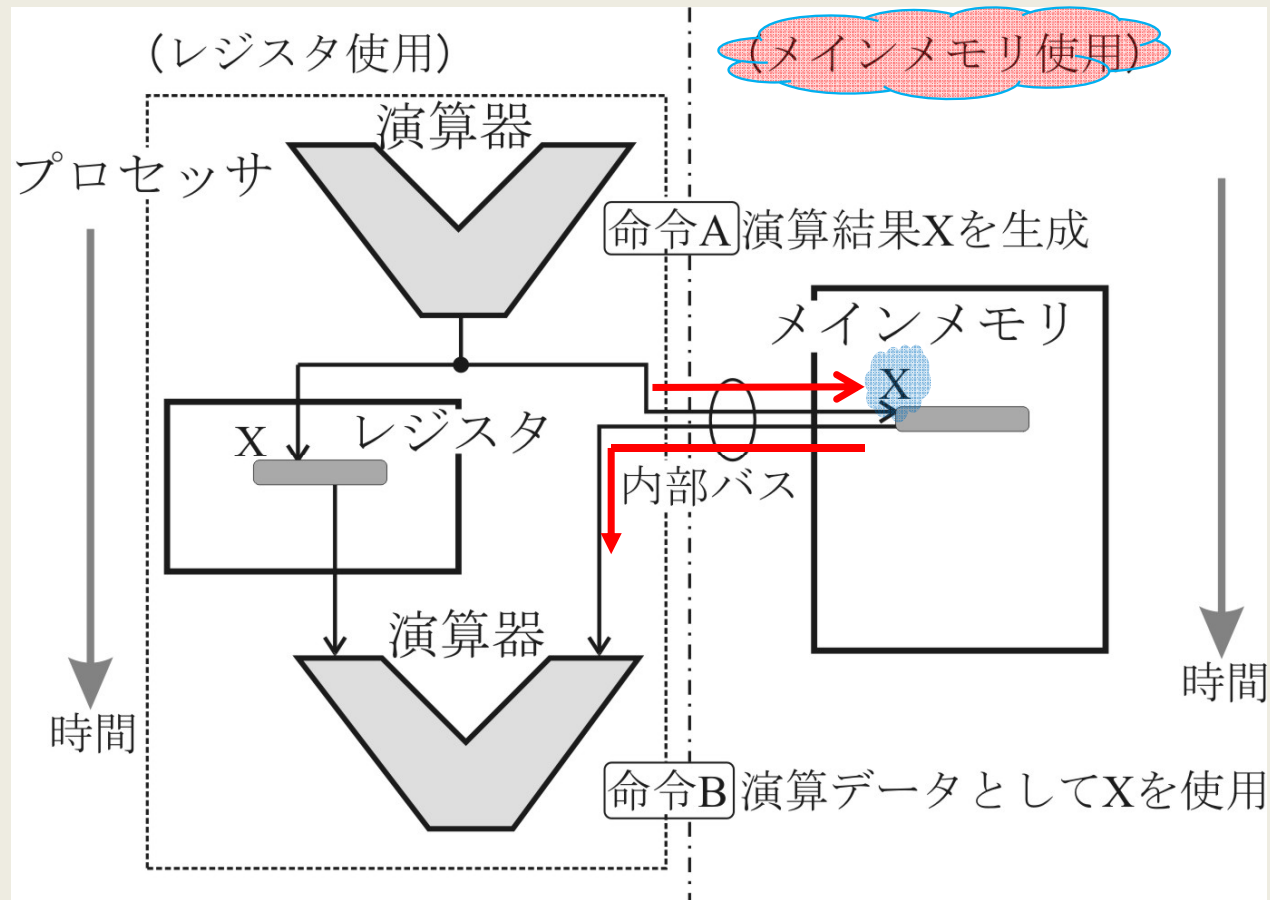
- AとBとの間で, Xの受け渡し

- AがXをメインメモリに書き込む操作 + BがXをメインメモリから読み出す操作

= プロセッサは都合2回メインメモリと通信する必要

↓ (このような無駄な動作を回避するために)

マシン命令間でのデータの共用(図)

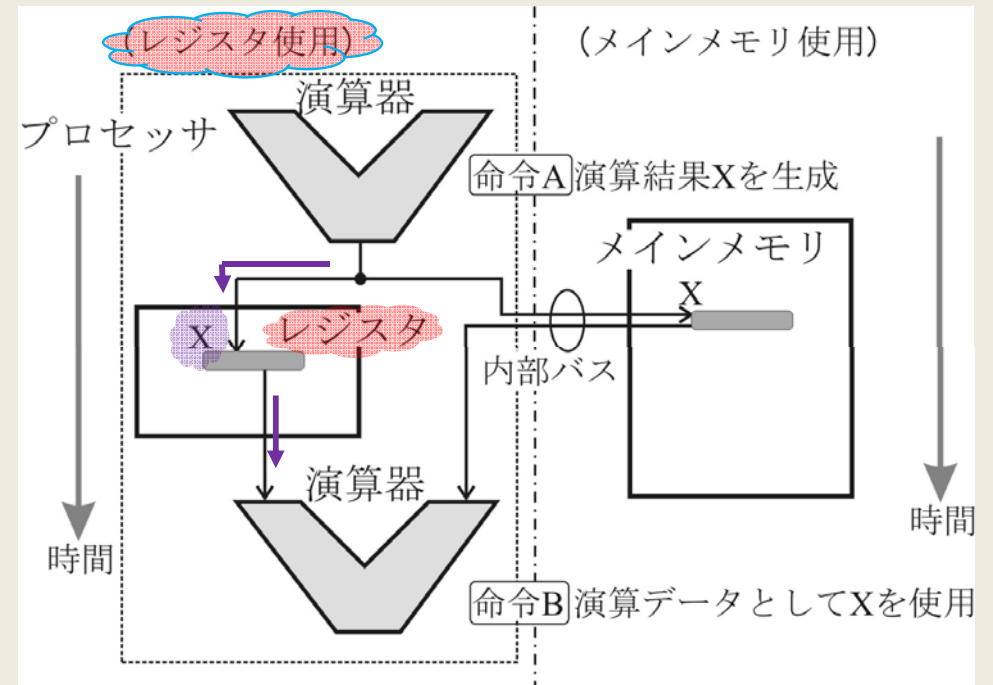


レジスタ(register)

- プロセッサ内に装備するハードウェア装置, メモリ機構

- プロセッサ内に用意する引き続く一連のマシン命令列で共用されるデータを一時的に格納しておく場所

= 汎用レジスタ



プロセッサのメモリ機構としてのレジスタ

➤ メインメモリに比べると、次の特徴を持つプロセッサ内のメモリ機構

- (1) プロセッサ内でのデータ移動であるので、内部バスを経由する(プロセッサ外と通信する)プロセッサ-メインメモリ間転送よりも動作速度は圧倒的に速い
- (2) プロセッサのハードウェア機構の一部を占有するので、容量(個数, 大きさ)は小さくする必要

レジスタの効果

- ソースオペランドや結果オペランドで指定するデータの格納場所として、メインメモリのほかにも、レジスタをあてることが可能に
- 一連のマシン命令列の実行中に、複数の命令間で共用されたり受け渡しされたりして、何回も読み書きされるデータは、できる限りレジスタに置く
 - 「できる限り」という意味 = 「レジスタの個数は少数なので、その全部を使い切るまで」
 - 使い切ってしまうと、「それらをやりくりする」必要

レジスタとメインメモリの使い分け

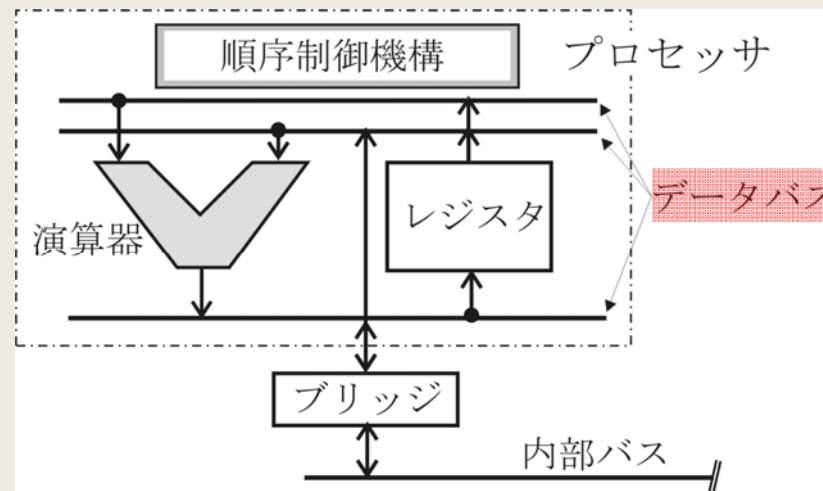
- (1) 一連のマシン命令列で使われる共用データは、レジスタ個数の範囲内で、**レジスタに一時的に置く**
- (2) 計算途中でなくなってしまうようなデータ、**最終的な計算(処理)結果などは、メインメモリに格納**

オペランドとしてのレジスタの指定

- **レジスタ**: レジスタ番号
- **メインメモリ**: メインメモリアドレス

データバス(data bus) (1)

- 主として演算器－レジスタ間のデータ転送専用使用するバス
- 内部バス (プロセッサ－メインメモリ間を結ぶ) : マシン命令とデータの両方の転送で共用



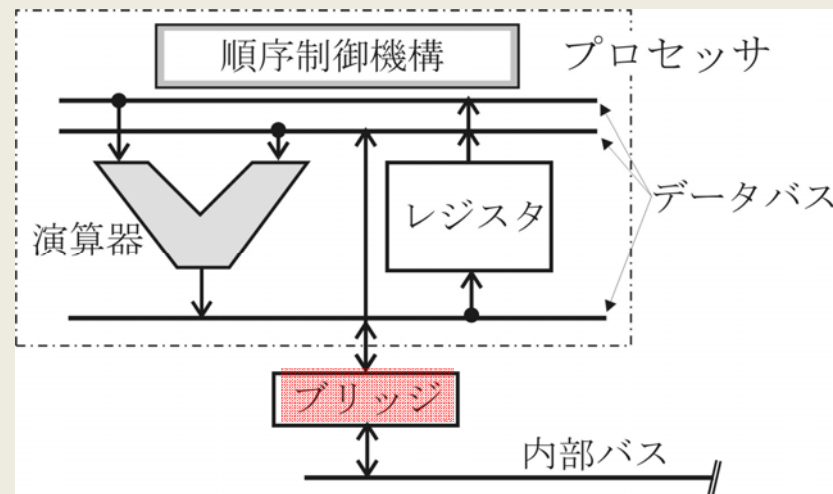
データバス(data bus) (2)

◆2項演算の場合

- ソースオペランドは(それがレジスタかメインメモリのいずれに格納されていようとも)同時に**2個**のデータが必要
 - 演算器の**2個**の**入口**に**別々**のデータバスがつながれている場合には, ソースオペランドの演算器への**同時転送(オペランド取り出し)**が可能
 - 演算器の2つの入り口への転送路を**単一**のデータバスで**共用**する場合 → **転送タイミングをずらす必要**
- **マイクロプロセッサ, 最新のプロセッサ**: 2本のソースオペランド用バスと1本の結果オペランド用バスとの計**3本**の**データバス**を別々に装備(前図)

ブリッジ(bridge)

- データバスと内部バスなどのように、**バスどうしを接続するハードウェア機構**
 - **ブリッジ**の主な役割: 転送(動作)速度や方式の相異なる**バス間のタイミング(timing)合わせ**や**調停**



(重要)

プログラムカウンタ(program counter; PC)

- 命令実行サイクルに入っている(実行中の)マシン命令のメインメモリアドレスを記憶
 - プロセッサに装備され, マシン命令を実行するためのハードウェア機構
 - 実行するマシン命令の順序制御 = PCの管理

命令実行サイクル(1)

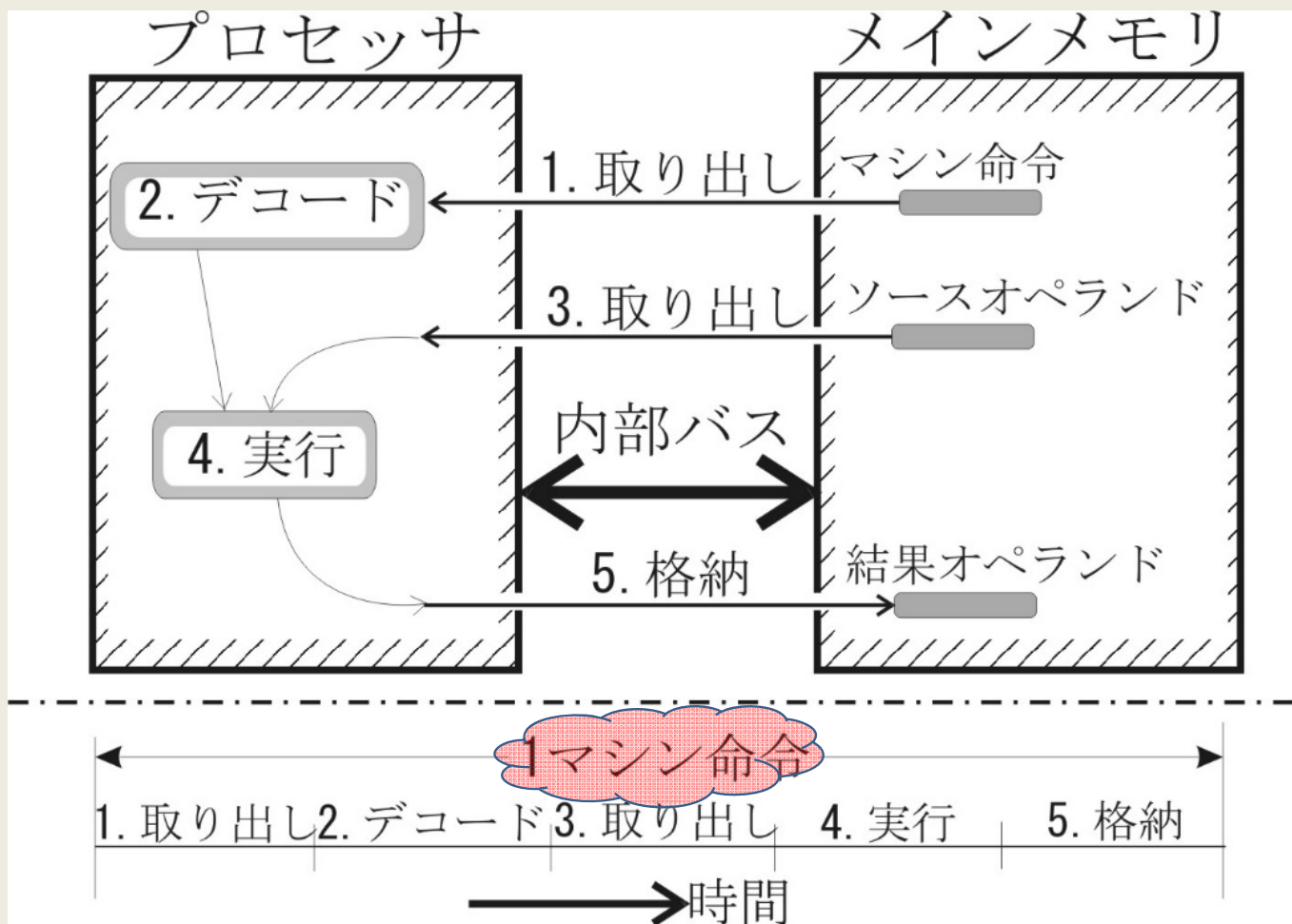
- **ステージ**(stage; 過程, 段階): マシン命令の各**部分機能**の実現過程
 - マシン命令の機能は**共用**する複数ステージによって実現
 - ステージごとに分担する機能は**相異なる**
 - ステージは**この順**で実行する必要

命令実行サイクル(7)

- 1～5の各ステージすべてをこの順で行って1マシン命令の実行が完了
- マシン命令列(マシン語プログラム)の実行は, この1～5のステージをマシン命令ごとにくり返す

■ **命令実行サイクル**(cycle): マシン命令機能(=1～5のステージ)の実現ごとのくり返し

命令実行サイクル(図)



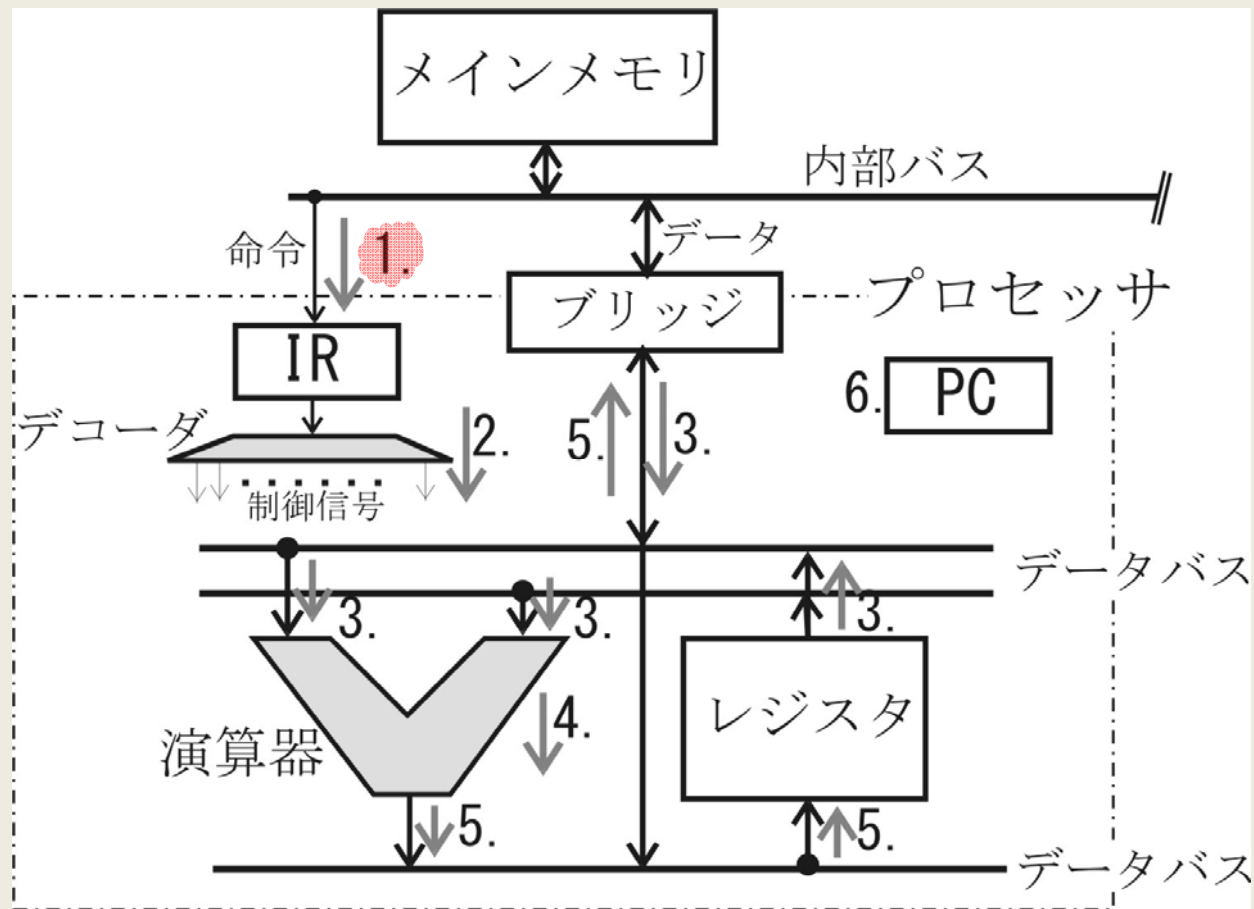
命令実行サイクルのステージ動作(1)

◆内部装置を構成するハードウェア機構間での命令やデータの移動を中心に

1. **命令取り出し**: 直前の命令によってPCに設定されたメインメモリアドレスに格納されている**マシン命令**をメインメモリから読み出し, 内部バスを經由して, プロセッサ内の**命令レジスタ(IR)**に置く

➤ 命令実行サイクル中はIRが現在実行中のマシン命令を保持

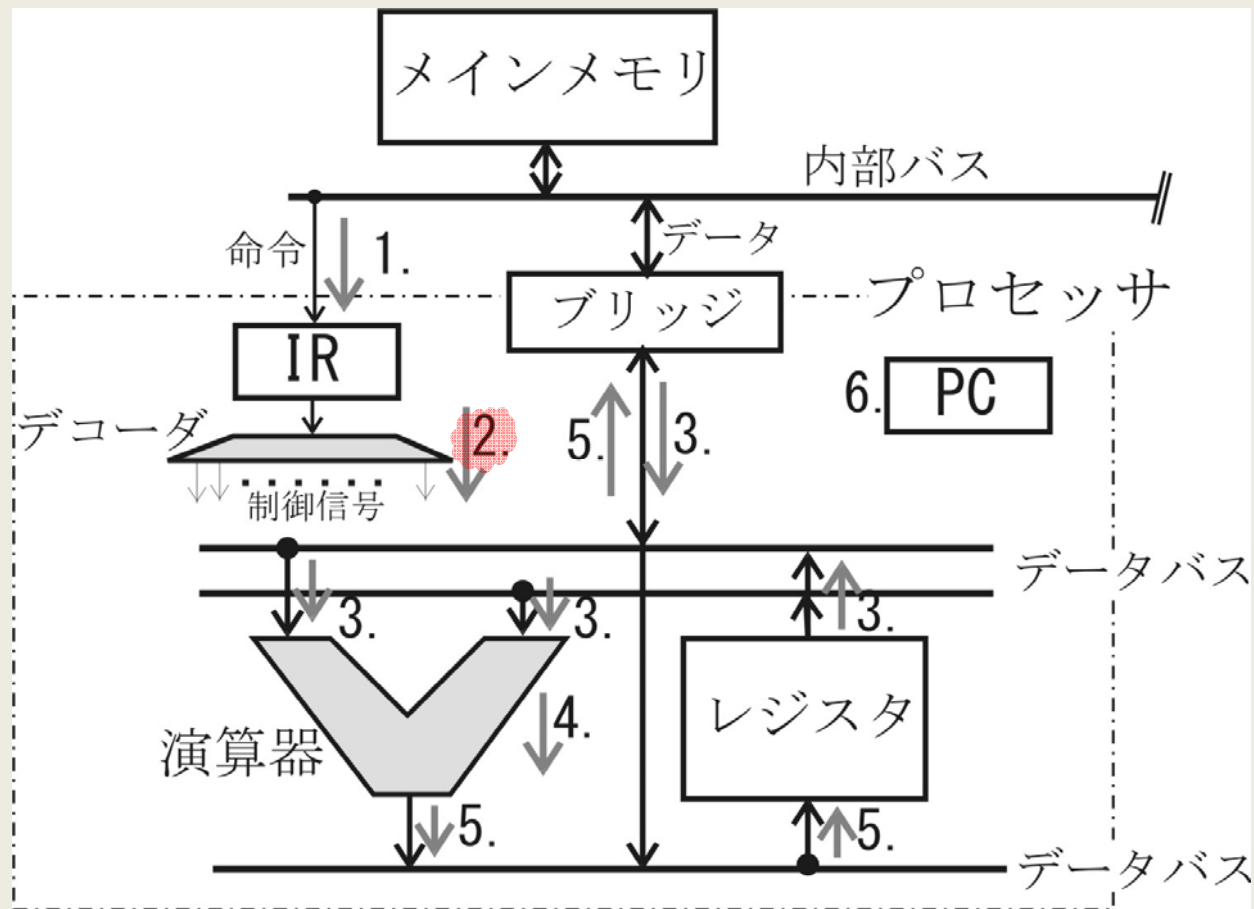
命令実行サイクルとプロセッサの動作(図)(1)



命令実行サイクルのステージ動作(2)

2. **命令デコード**: IRから**マシン命令**を取り出し, **デコーダ** (decoder; 復号器)を通して, マシン命令中に符号化され埋め込まれていた**命令コード**や**オペランド**などの各情報ごとに分離して取り出し, 実際の**制御信号**などに展開(デコード)して必要なハードウェア機構に**分配**

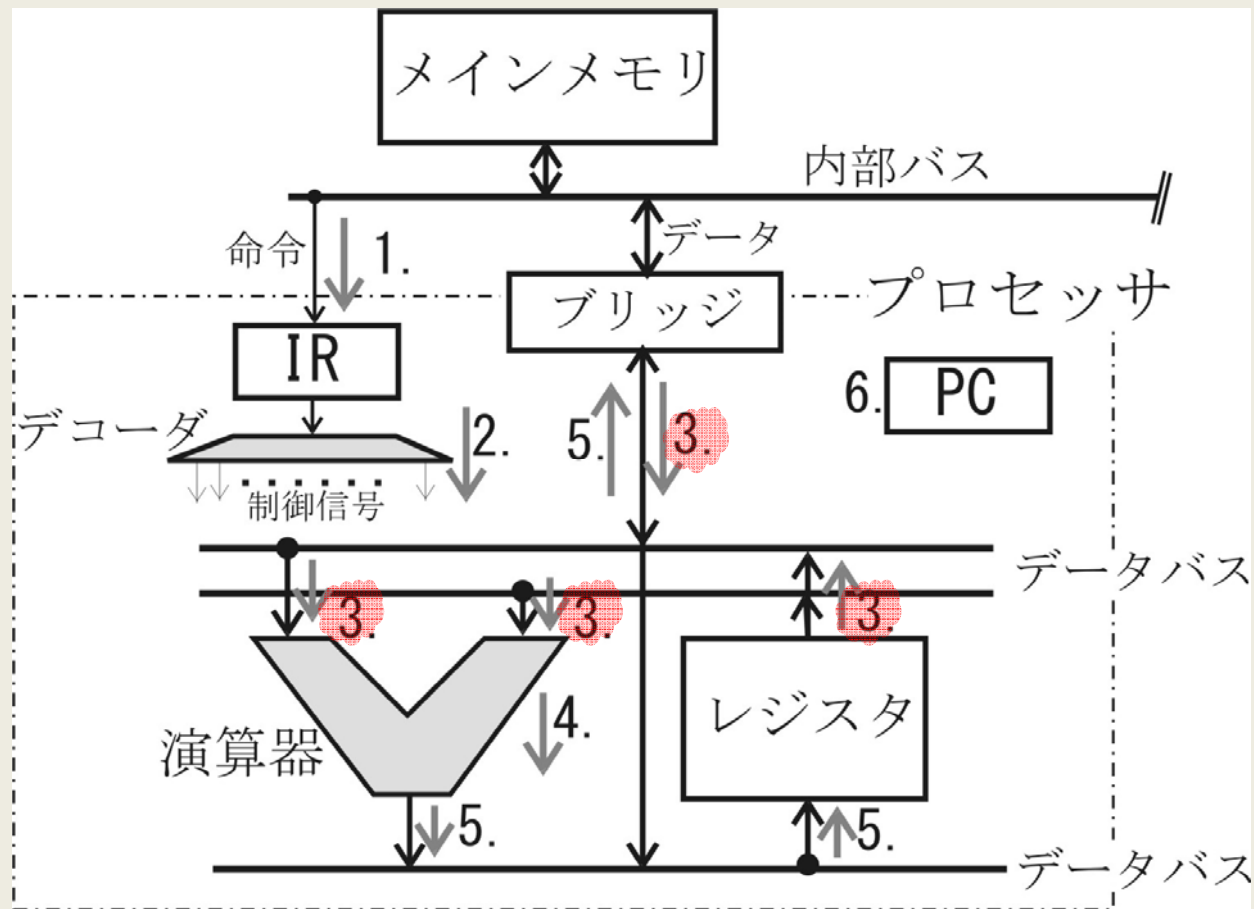
命令実行サイクルとプロセッサの動作(図)(2)



命令実行サイクルのステージ動作(3)

3. **オペランド取り出し**: デコーダによって展開されたソースオペランドの指定によって, **レジスタ**あるいは**メインメモリ**から演算に使われる**データ**を取り出し, 内部バスやデータバスを経由して**演算器の入口**に置く
- 2項演算ならば, 演算に必要なデータは2個であり, このステージを2回行う必要
 - このステージ用のハードウェア機構(**データバス**など)が2組以上あれば同一ステージで同時に可能

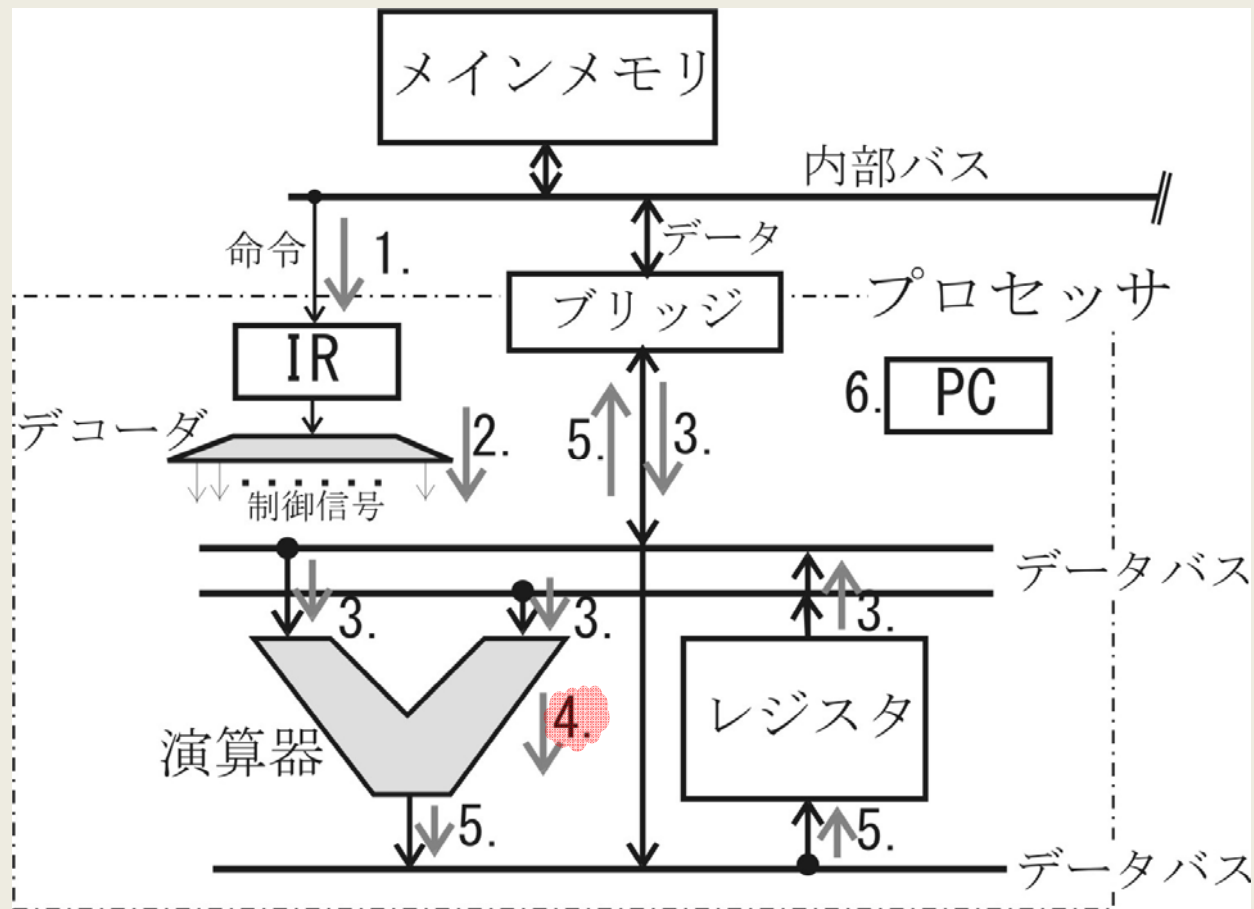
命令実行サイクルとプロセッサの動作(図)(3)



命令実行サイクルのステージ動作(4)

4. **実行**: **演算器**による**演算**を**入口**に置かれたデータを用いて行い, その**出口**に置く
- 演算種類によって使用される演算器は異なる
 - **順序制御命令**のように, **演算器を使用しない**実行機能もある

命令実行サイクルとプロセッサの動作(図)(4)



命令実行サイクルのステージ動作(5)

5. **結果格納**: 演算器の出口に置かれた**演算結果データ**を, 2でデコードされた結果オペランドの指定にしたがって, データバスや内部バスを經由して**レジスタ**あるいは**メインメモリ**に書き込む

命令実行サイクルとプロセッサの動作(図)(5)

