

## 第2章 ノイマン型コンピュータ

---

---

## 2.1 ノイマン型コンピュータの基本構成

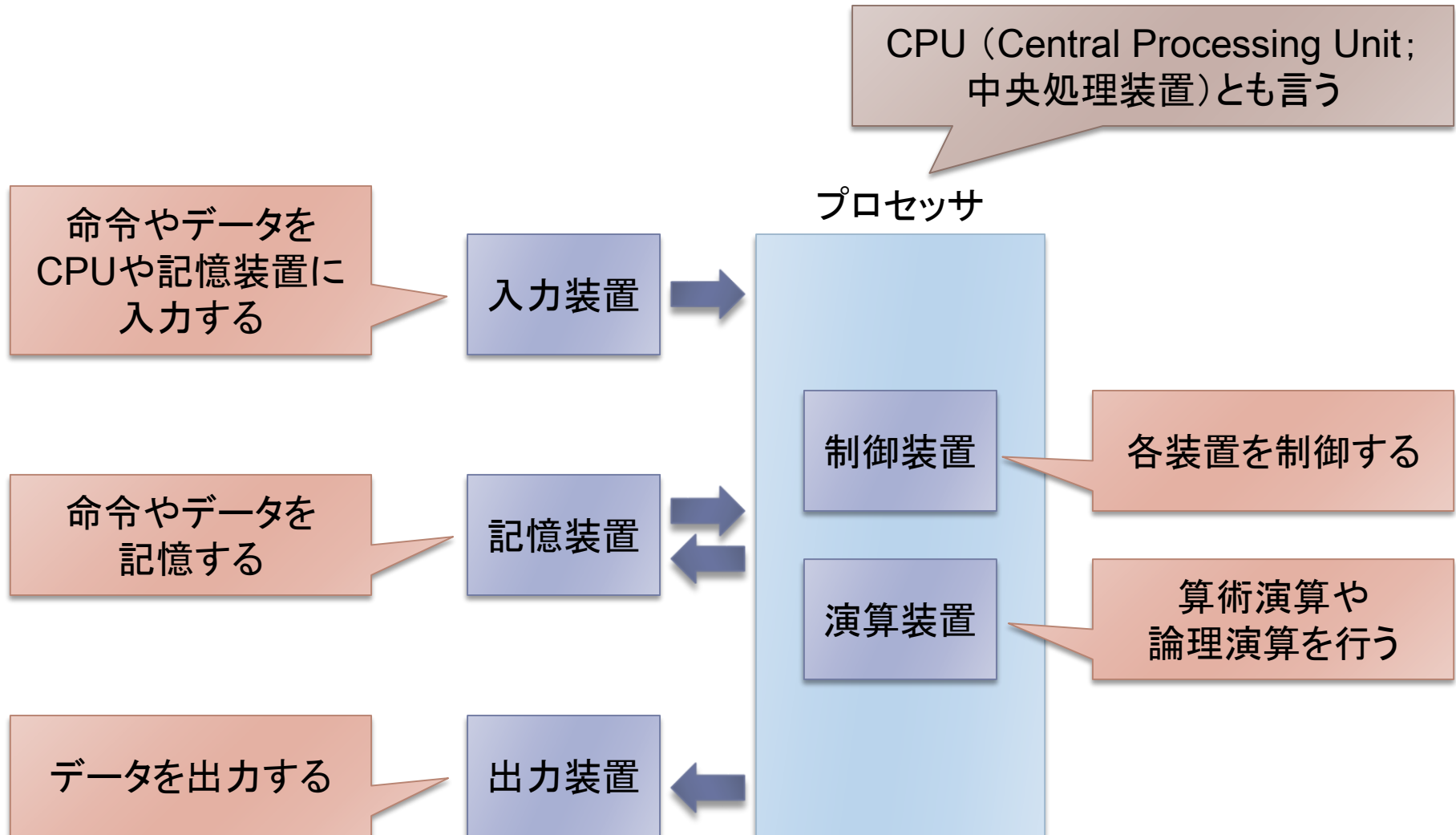
---

# ノイマン型コンピュータ

---

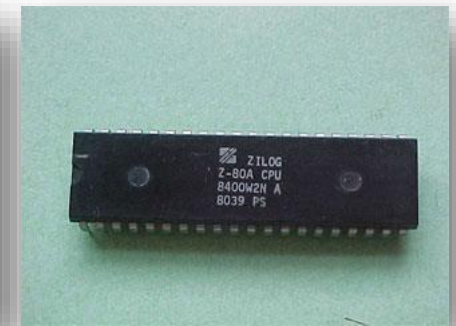
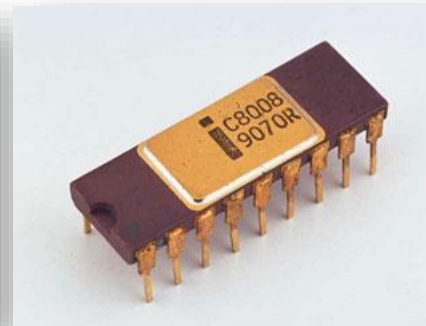
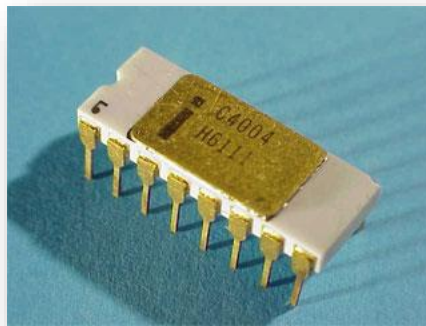
- ▶ ノイマン型コンピュータ
  - ▶ フォンノイマンらが提示したコンピュータの構成方式に従うコンピュータを、**ノイマン型コンピュータ**という。
  - ▶ 現在のコンピュータの主流は、ノイマン型コンピュータである。
- ▶ ノイマン型コンピュータの特徴
  - ▶ **プログラム(可変)内蔵方式**
    - ▶ 実行するプログラムや処理するデータをメモリに格納しておき、実行時に必要なプログラムを読み出す。
  - ▶ **逐次処理方式**
    - ▶ メモリに格納された命令を、順次読み出しながら処理を進める。
  - ▶ **単一メモリ方式**
    - ▶ プログラムとデータは、同じメモリ内に格納されている。

# ノイマン型コンピュータの基本構成



## CPUの発展（初期）

年代	1971	1974	1976
型名	インテル 4004	インテル 8080 モトローラ 6800	インテル 8085 モトローラ 6809 ザイログ Z80
ビット幅	4ビット	8ビット	8ビット
トランジスタ数	2300	8500	1万
クロック周波数	750 kHz	1 MHz	5 MHz



## CPUの発展（現代）

年代	1993	2003	2010
型名	インテル Pentium モトローラ他 PowerPC	インテル Pentium 4	インテル Core i7
ビット幅	32ビット	32ビット	64ビット
トランジスタ数	310万	1億2500万	7億以上
クロック周波数	100 MHz	3 GHz以上	3 GHz以上

（注）CPUの複雑化に伴い、単純に“ビット幅”と呼ぶことができなくなりつつある。



# CPUの多様化

## ▶ CPUの多様化

- ▶ 現在では、用途ごとに、数多くのCPUが開発されている。

用途	メーカー	型名
パソコン	インテル	Core 2 Duo Core i7/i5/i3
	AMD	Athlon(アスロン) Phenom(フェノム)
モバイルパソコン	インテル	Atom(アトム) Core i7/i5/i3
	AMD	Turion(テュリオン)
ワークステーション	インテル	Xeon(ジーオン) Itanium(アイテニウム)
組み込み	ルネサスエレクトロニクス (日本電気, 日立製作所, 三菱電機)	H8

# 演習問題

---

## ▶ 問題1

- ▶ ノイマン型コンピュータとは, どのようなコンピュータのことを言うのか, 特徴を挙げて説明せよ.

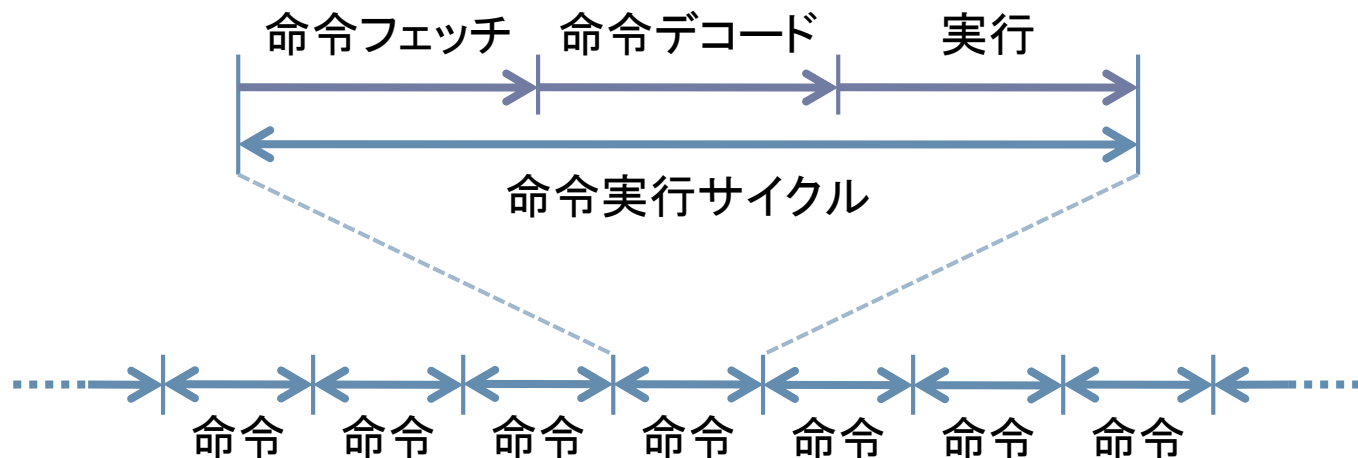


## 2.2 ノイマン型コンピュータの基本動作

---

# 命令実行サイクル

- ▶ ノイマン型コンピュータでは,
  - ▶ メインメモリに格納されている命令を取り出す. (命令フェッチ)
  - ▶ 取り出した命令を解読する. (命令デコード)
  - ▶ 解読した情報に基づいて, 所定の処理を実行する. (実行)という動作を繰り返して行うことにより, 命令が順次実行される.
- ▶ 1つの命令が実行される一連の流れを, **命令実行サイクル**という.

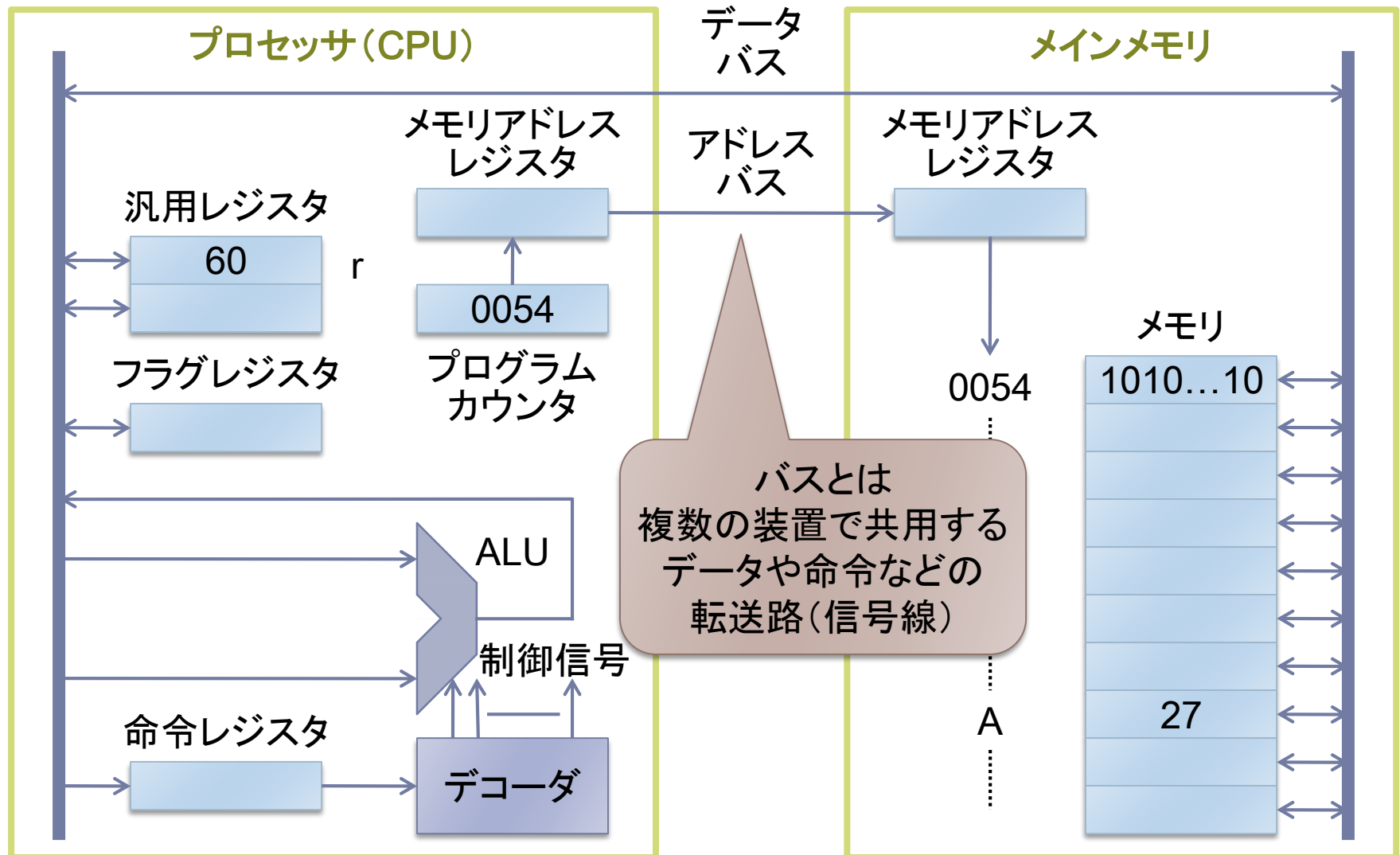


# CPUの構成

---

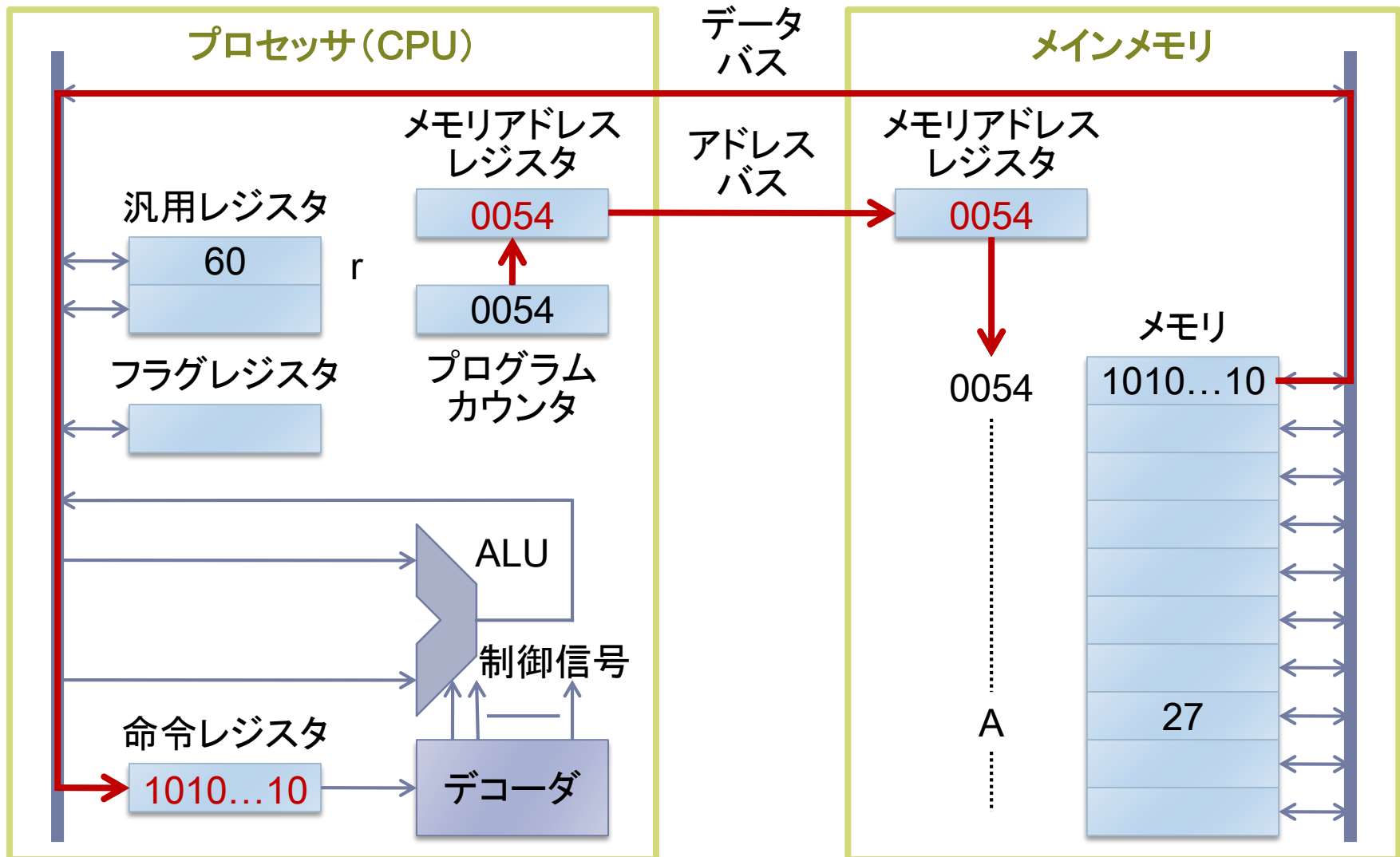
- ▶ 演算装置
  - ▶ 算術演算装置 (ALU ; Arithmetic and Logic Unit)
    - ▶ 算術演算または論理演算を行う演算回路.
  - ▶ 汎用レジスタ
    - ▶ データを一時的に記憶するメモリ. 高速に動作する.
  - ▶ フラグレジスタ
    - ▶ 演算命令の実行などによって値が設定されるメモリ.
- ▶ 制御装置
  - ▶ プログラムカウンタ
    - ▶ “次に実行する命令が格納されているメモリアドレス”を記憶するメモリ.
  - ▶ 命令レジスタ
    - ▶ メモリから取り出された命令を, 一時的に記憶するメモリ.
  - ▶ デコーダ
    - ▶ 命令レジスタに記憶されている命令を復号して, 命令の実行に必要な制御信号を出力する.

# プロセッサとメインメモリ（概略構成）



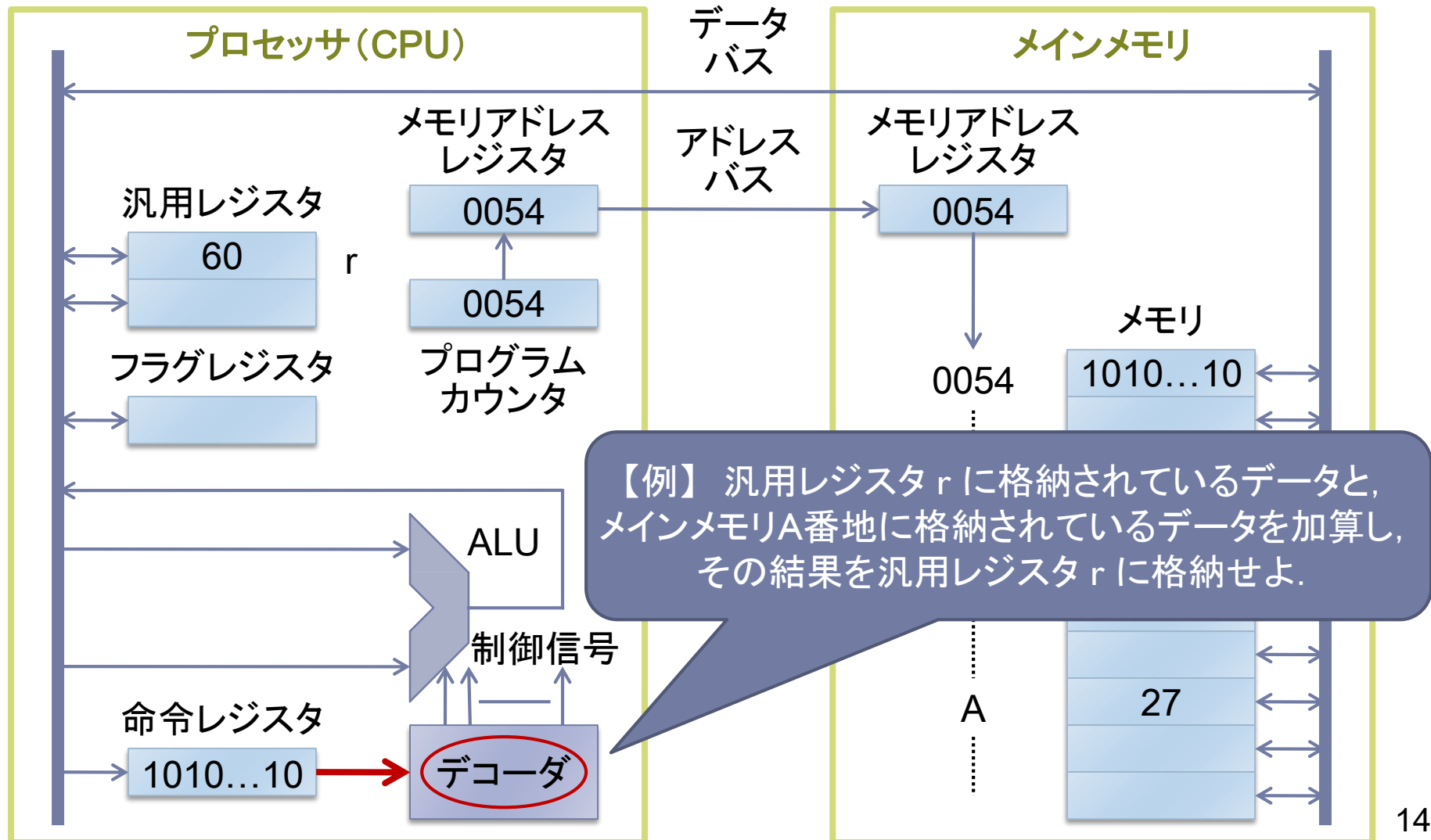
# 命令実行サイクル

## 命令フェッチ

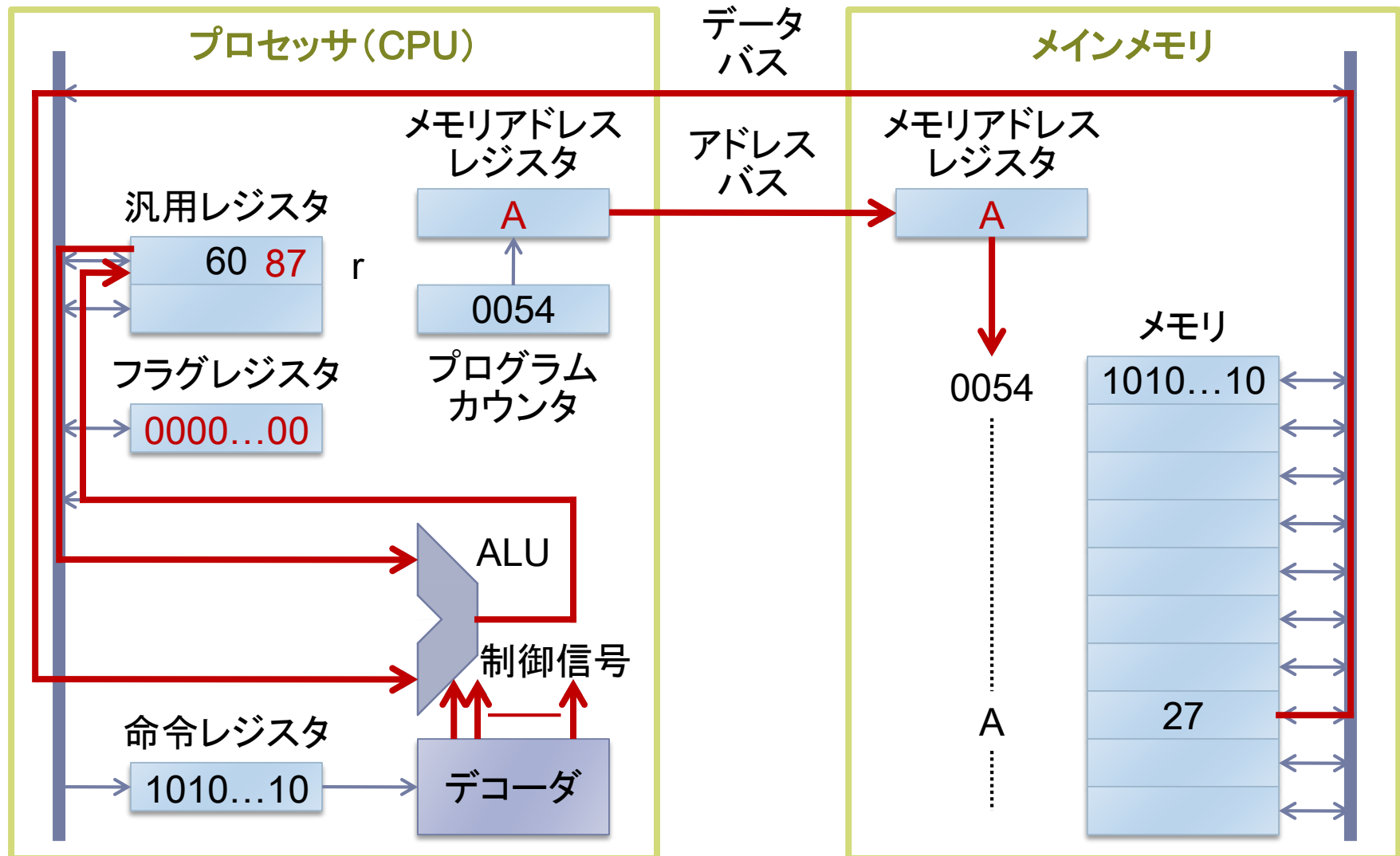


# 命令実行サイクル

## 命令デコード

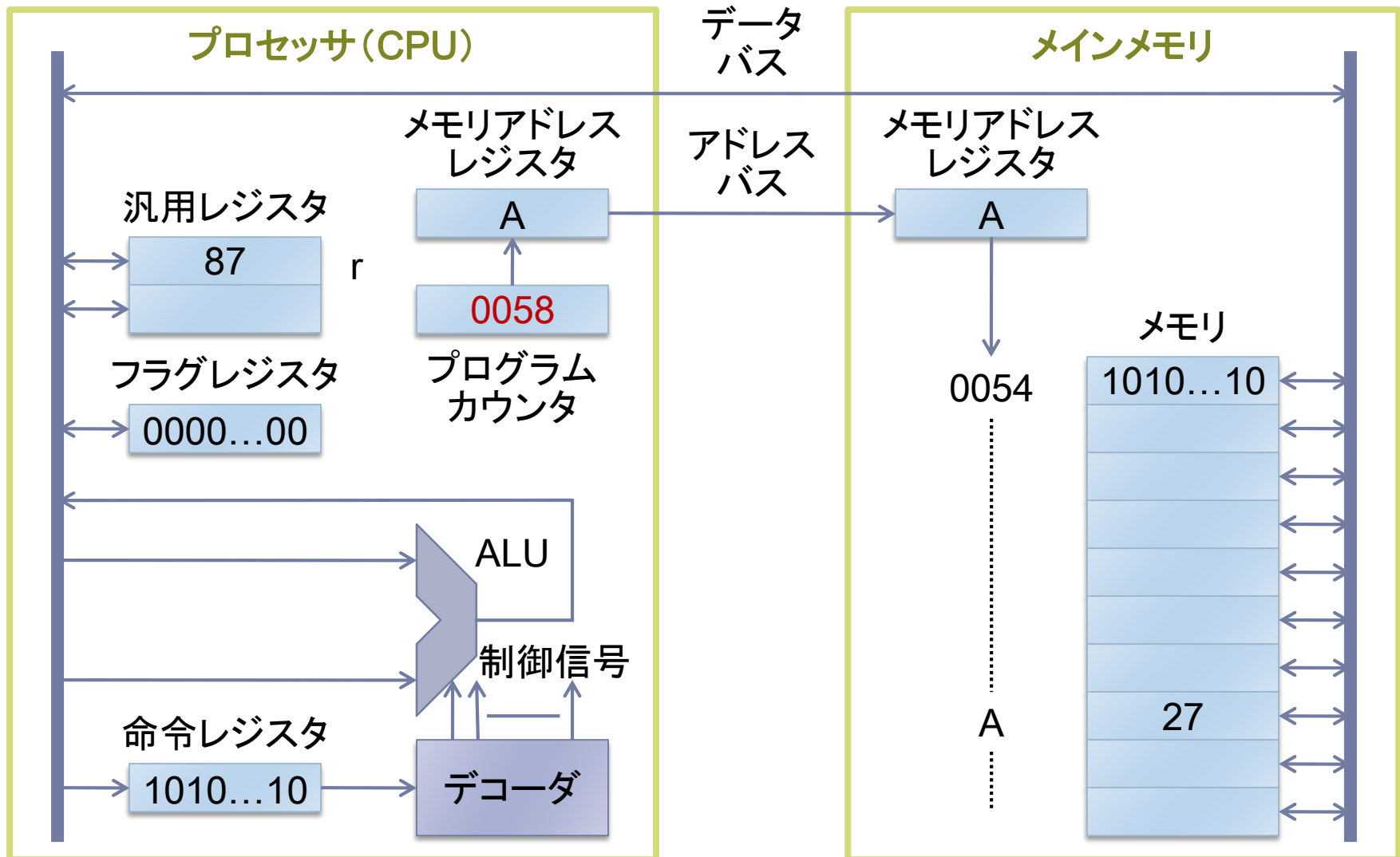


# 命令実行サイクル 実行



# 命令実行サイクル

## 次命令アドレス決定





# フォンノイマンボトルネック

---

- ▶ フォンノイマンボトルネック
  - ▶ ノイマン型コンピュータでは、同じメモリに命令とデータが格納されている。
  - ▶ また、プロセッサとメインメモリは、バスと呼ばれる転送路を通じて、命令やデータの転送を行っている。
  - ▶ そのため、「命令の取り出し」と「データの転送」が、バスの使用权をめぐって競合することになり、この部分の性能がコンピュータ全体の性能左右することが多い。
  - ▶ このような「プロセッサ－メモリ間の転送性能がコンピュータ全体の性能を左右する」という構造的な問題点を、**フォンノイマンボトルネック**という。

# 演習問題

---

## ▶ 問題2

- ▶ ノイマン型コンピュータでは、「命令フェッチ」、「命令デコード」、「実行」という一連の流れに従い、1つの命令が実行される。また、次の命令を実行するための「次命令の決定」が行われる。
- ▶ 以下の(a)～(h)に適切な語句を入れよ。
- ▶ 「命令フェッチ」では、
  - ▶ プロセッサが、(a)に格納されているアドレスを、(b)に出力する。
  - ▶ メインメモリは、プロセッサが(b)に出力したアドレスを受け取り、そのアドレスに格納されている(c)を、(d)に出力する。
  - ▶ プロセッサは、メインメモリが(d)に出力した(c)を、(e)に受け取る。

(次ページに続く)

# 演習問題

---

## ▶ 問題2（続き）

- ▶ 「命令デコード」では,
  - ▶ プロセッサは, (e)に受け取った(c)を(f)に送り, (g)する.
- ▶ 「実行」では,
  - ▶ プロセッサは, (g)結果に基づいて, (h)を生成する. その(h)に応じて, プロセッサ内のデータ転送, プロセッサ-メインメモリ間のデータ転送, 所望のデータに対する演算などを行う.
- ▶ 「次命令の決定」では,
  - ▶ 次に実行する命令が格納されているアドレスを, (a)に設定する.

# 演習問題

---

## ▶ 問題3

- ▶ フォンノイマンボトルネックとは, どのような問題点のことを言うのか, 説明せよ.

## 第3章 命令セットアーキテクチャ

---

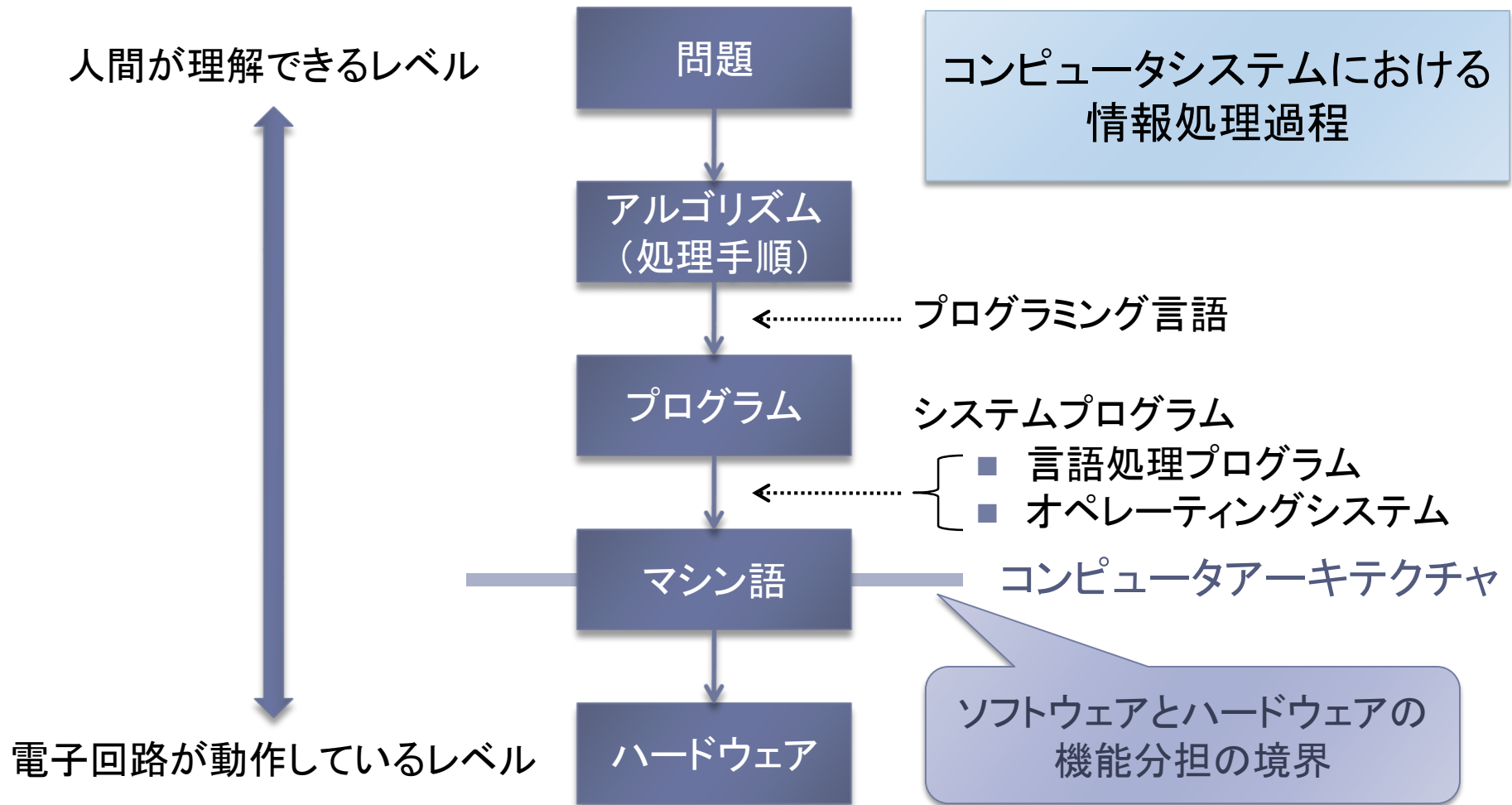
---

## 3.1 命令

---

# [Review]

## コンピュータシステムにおける情報処理過程

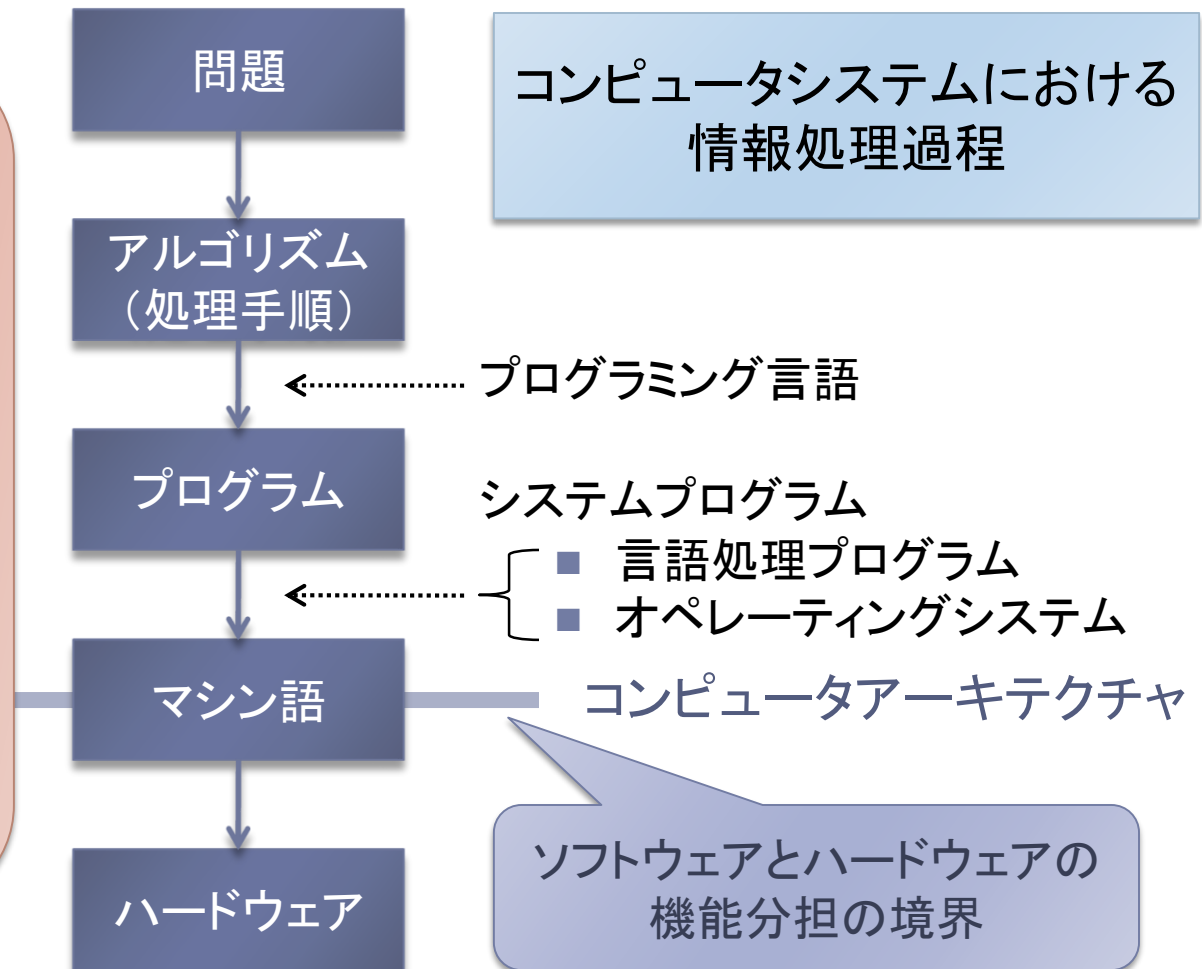


# 機械語命令（マシン命令）

コンピュータハードウェアに直接的に動作を指示するのは、マシン語である。

マシン語に基づいて生成された制御信号が、ハードウェアに所望の動作をさせる。

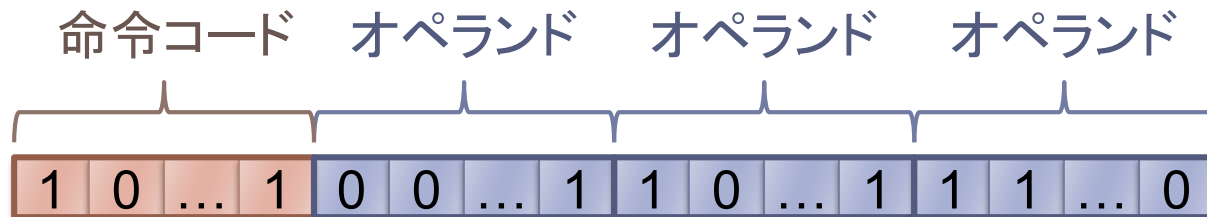
マシン語は、マシン命令、機械語命令、単に命令とも呼ばれる。





# 命令の形式

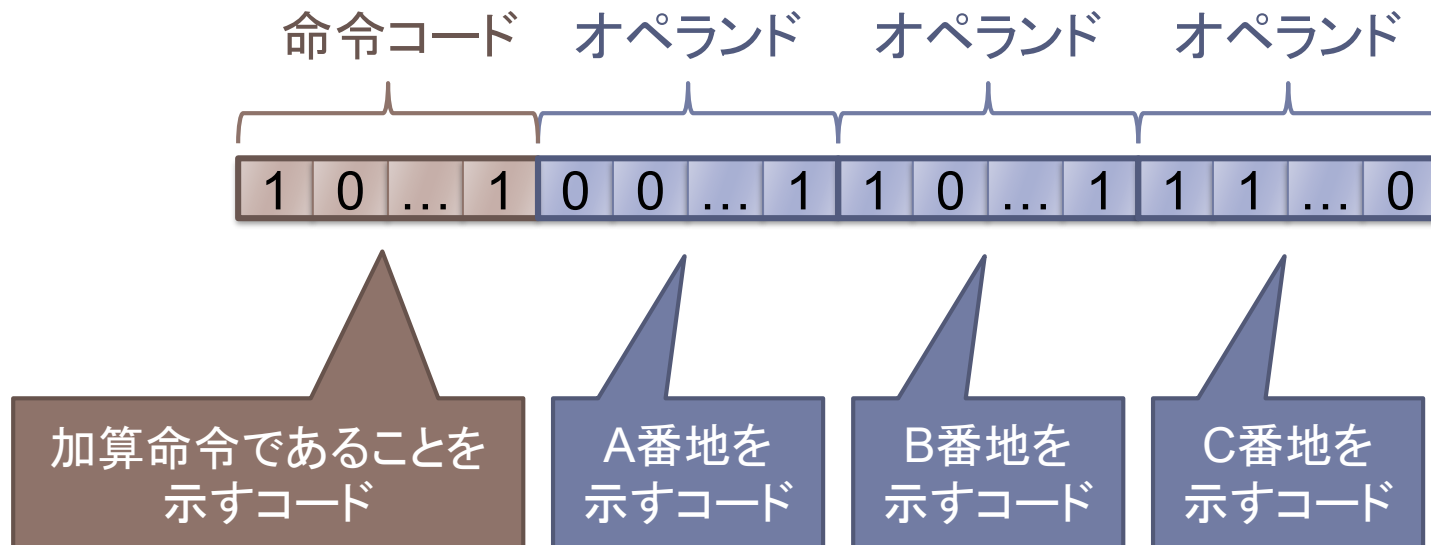
- ▶ 命令は、**命令コード**と**オペランド**により構成される.
  - ▶ **命令コード**(operation code)
    - ▶ 命令の種類を示す.
    - ▶ 演算コード, OPコード(オペコード)ともいう.
  - ▶ **オペランド**(operand)
    - ▶ 命令で使用するデータやデータの格納場所(アドレス)を示す.
    - ▶ 命令コードに従って, 1命令に0個以上を備える.



## 命令（例）

### ▶ 【例】 加算命令の例

- ▶ メインメモリのA番地に格納されている値と、メインメモリのB番地に格納されている値を加算し、加算結果をメインメモリのC番地に格納する.



# オペランドの種類

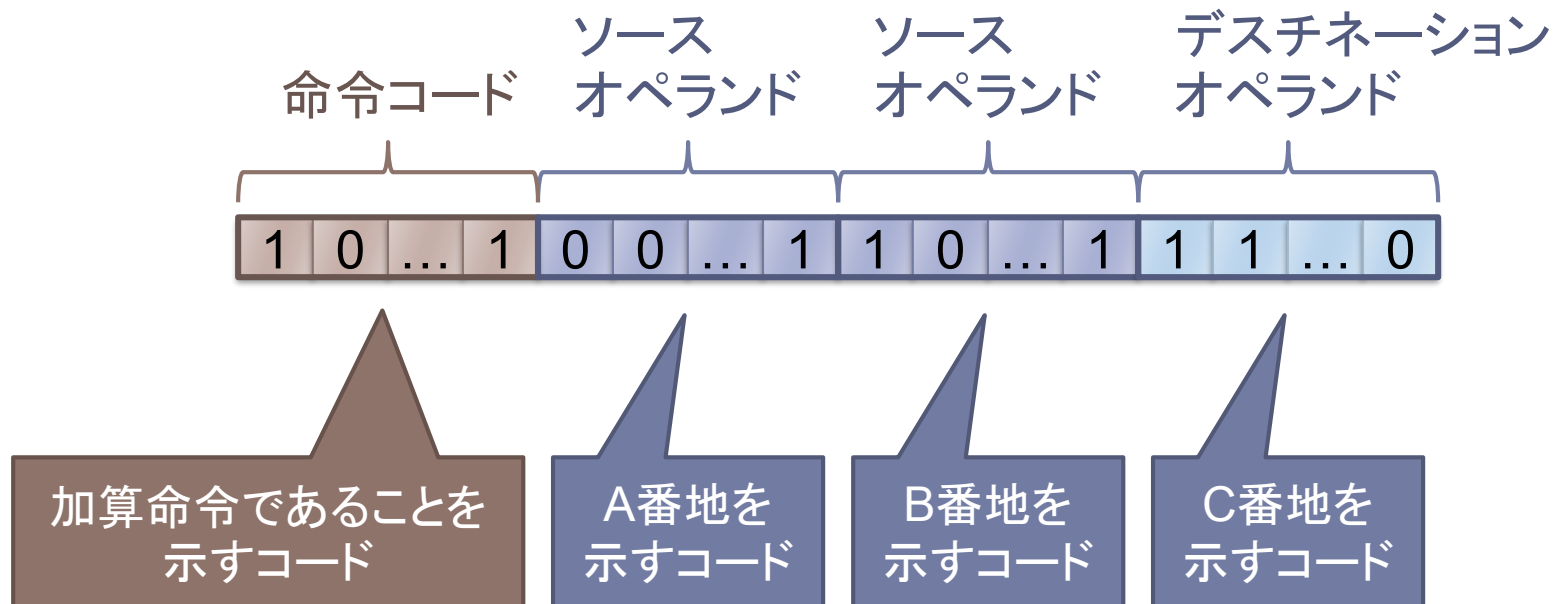
---

- ▶ オペランドの種類
  - ▶ オペランドには、ソースオペランドとデスチネーションオペランドがある。
    - ▶ ソースオペランド (source operand)
      - 処理するデータの格納元 (アドレスなど) を示す.
    - ▶ デスチネーションオペランド (destination operand)
      - 処理した結果データの格納先 (アドレスなど) を示す.

## オペランドの種類（例）

### ▶ 【例】 加算命令の例

- ▶ メインメモリのA番地に格納されている値と、メインメモリのB番地に格納されている値を加算し、加算結果をメインメモリのC番地に格納する。



## オペランド数による命令の分類

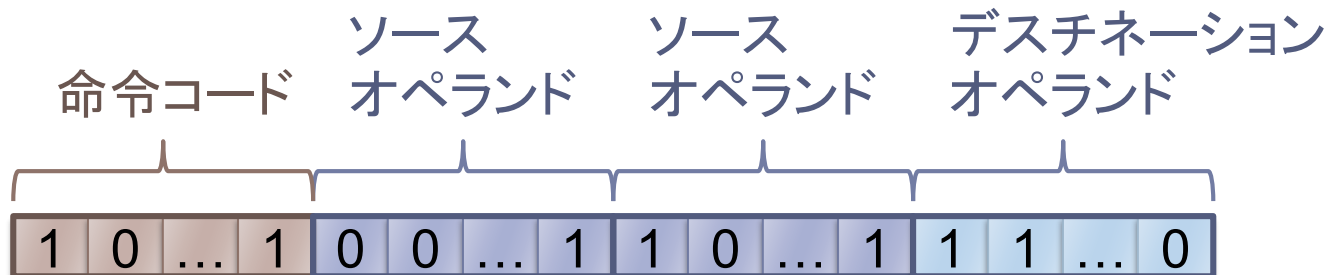
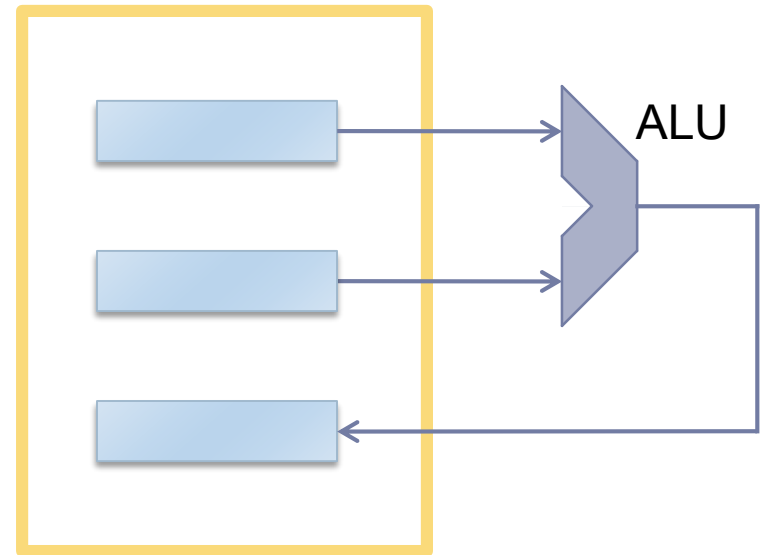
---

- ▶ オペランドの個数は、命令によって異なる.
- ▶ 命令は、オペランドの個数によって、以下のように分類することができる.
  - ▶ 3アドレス命令
  - ▶ 2アドレス命令
  - ▶ 1アドレス命令
  - ▶ 0アドレス命令

## 3アドレス命令

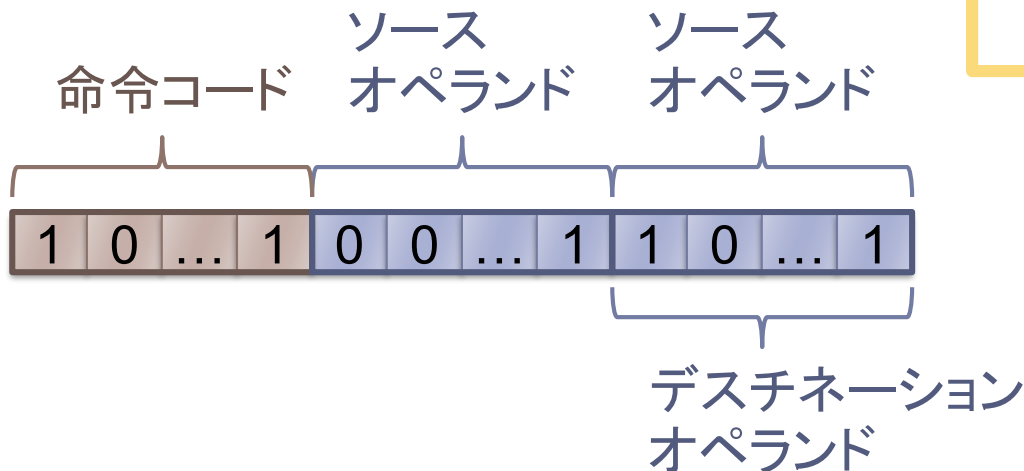
- ▶ 3アドレス命令
  - ▶ ソースオペランド2個と、デスチネーションオペランド1個を、そのまま記述する命令.
  - ▶ 考え方は簡単だが、命令が長くなるという欠点がある.

メインメモリ or レジスタ

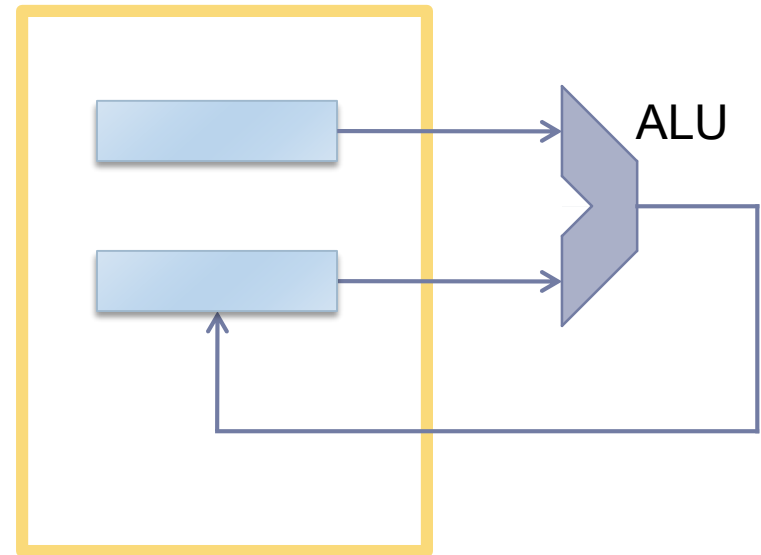


## 2アドレス命令

- ▶ 2アドレス命令
  - ▶ ソースオペランドのどちらか1個と、デスチネーションオペランドを、兼用する命令.
  - ▶ 使用するオペランドを1個節約できるが、兼用したソースオペランドに格納されていたデータは、上書きされることになる.

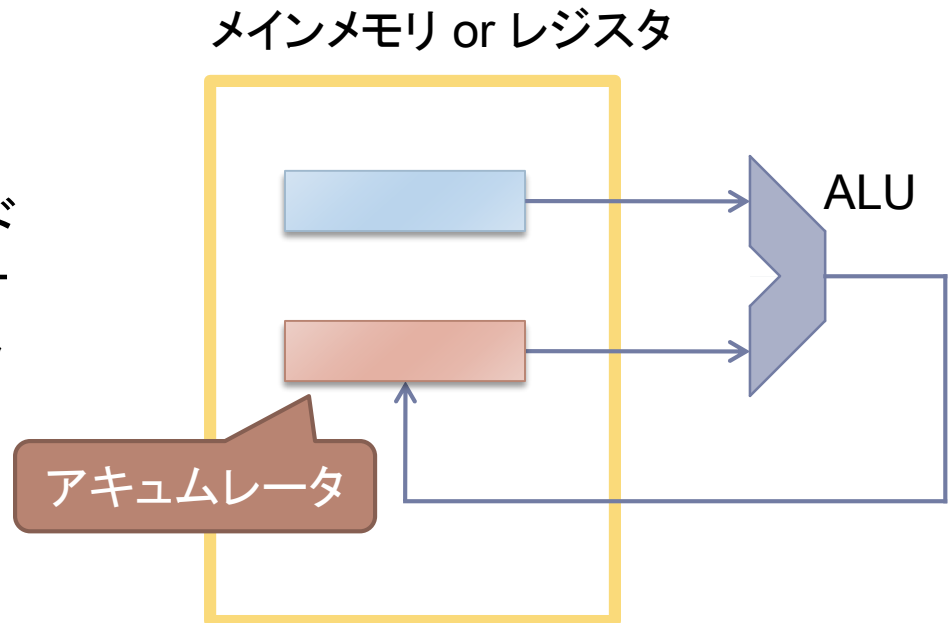
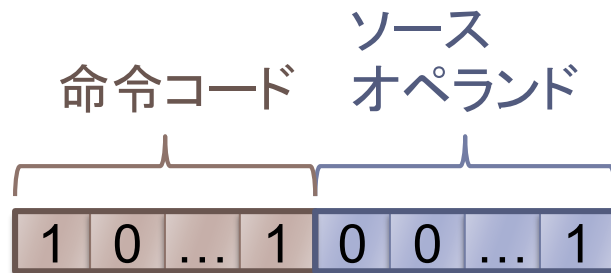


メインメモリ or レジスタ



# 1アドレス命令

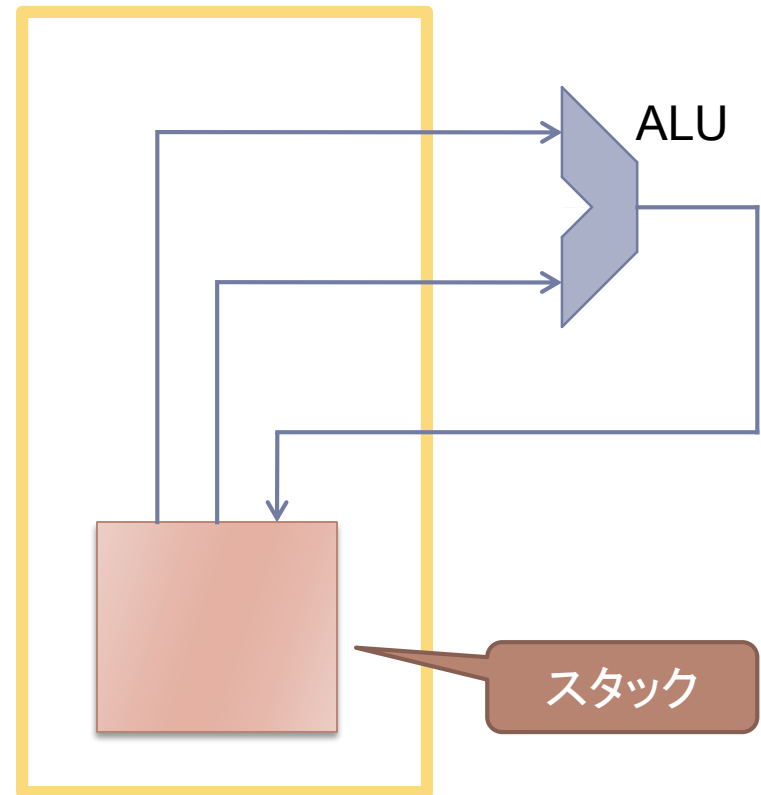
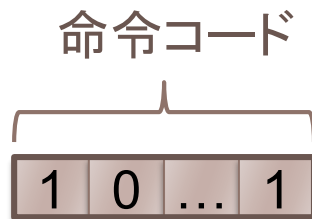
- ▶ 1アドレス命令
  - ▶ アキュムレータと呼ばれる特別な格納領域を使用する命令.
  - ▶ 命令には, 1個のソースオペランドのみを使用し, もう1個のソースオペランドとデスチネーションオペランドはアキュムレータとする.





## 0アドレス命令

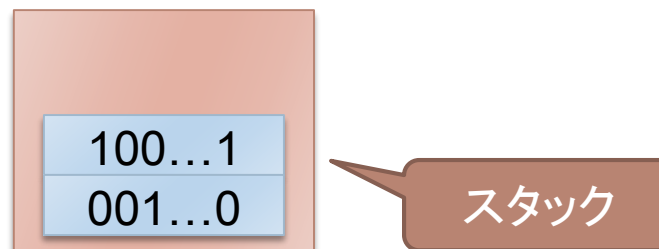
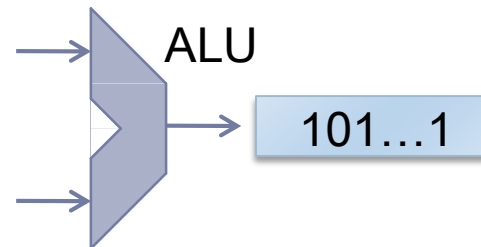
- ▶ 0アドレス命令
  - ▶ スタックと呼ばれる特別な格納領域を使用する命令.
  - ▶ 命令には, 命令コードのみを記述し, オペランドは記述しない.



# スタック

## ▶ スタック

- ▶ 保存するデータを順に積み重ねて記憶し、データを取り出すときには上から順に取り出す.
- ▶ スタックにデータを入れることを**プッシュダウン**または**プッシュ**, スタックからデータを取り出すことを**ポップアップ**あるいは**ポップ**という.



# 固定長命令方式と可変長命令方式

---

## ▶ 固定長命令方式

- ▶ すべての命令の長さ(ビット数)が一定である.
- ▶ 通常, 短い命令長を採用するため, ハードウェアを簡略化することができる.
- ▶ 一方で, 複雑な処理を実行する場合には, 多くの命令を組み合わせて使用する必要がある.

## ▶ 可変長命令方式

- ▶ 命令によって長さ(ビット数)が異なる.
- ▶ ハードウェアが複雑になる.
- ▶ 一方で, 1個の命令で多くの動作を指定できる.

# 演習問題

---

- ▶ 問題4
  - ▶ 以下の語句について説明せよ.
    - ▶ 命令コード
    - ▶ オペランド
    - ▶ ソースオペランド
    - ▶ デスチネーションオペランド
    - ▶ 3アドレス命令
    - ▶ 2アドレス命令
    - ▶ 1アドレス命令
    - ▶ 0アドレス命令