コンピュータ工学 I

― 中央処理装置 ―

Rev. 2019.01.07

コンピュータの基本構成とCPU

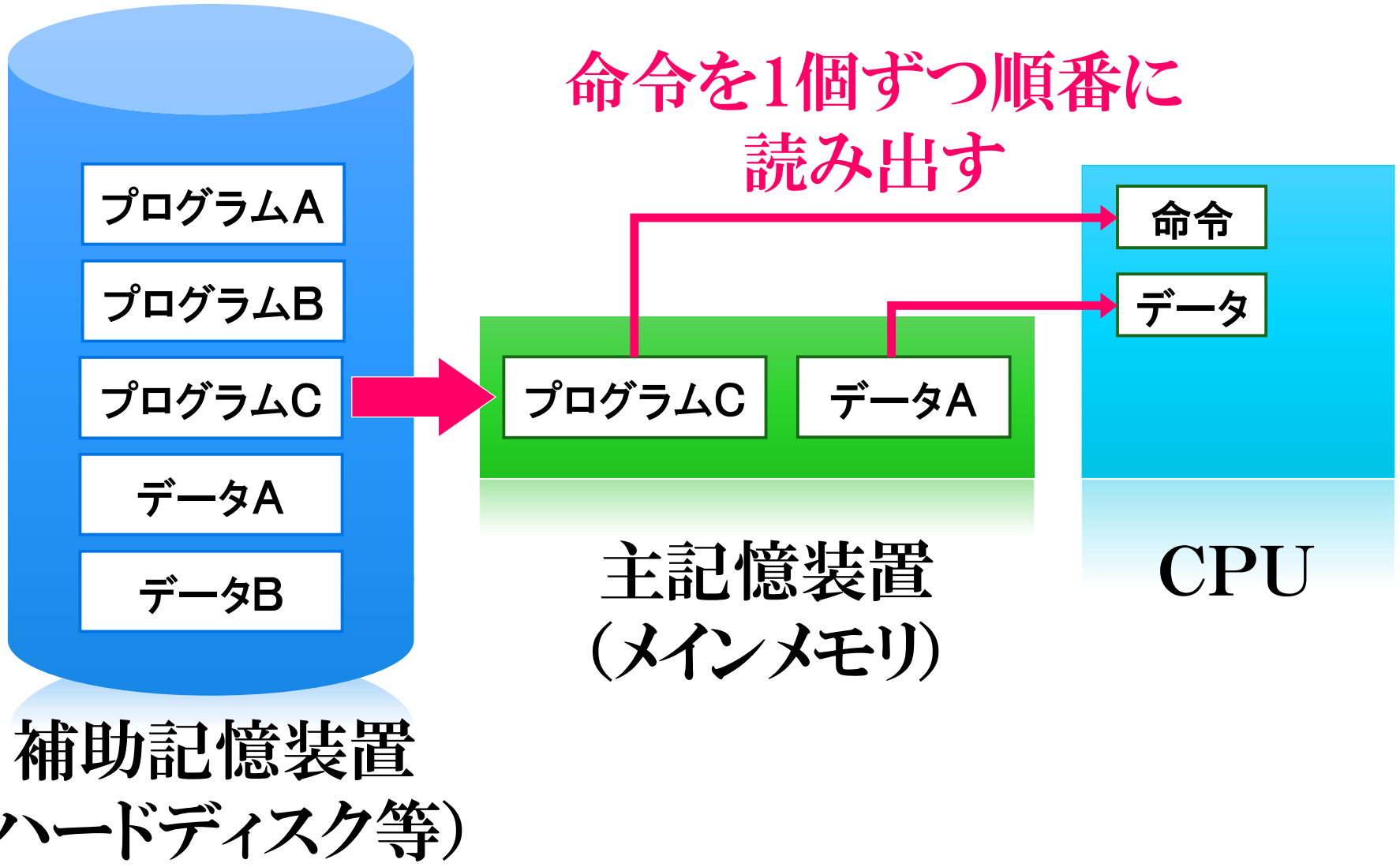
✣ 内容

1. CPUの構成要素
2. 命令サイクル
3. アセンブリ言語
4. アドレッシング方式
5. CPUの高速化
6. CPUの性能評価

コンピュータの構成装置

- **中央処理装置(CPU)**
  主記憶装置から命令を読み込み、実行を行う。
- **主記憶装置**
  CPUで実行するプログラム(命令の集合)やデータを記憶する。
- **補助記憶装置**
- **入力装置**
- **出力装置**

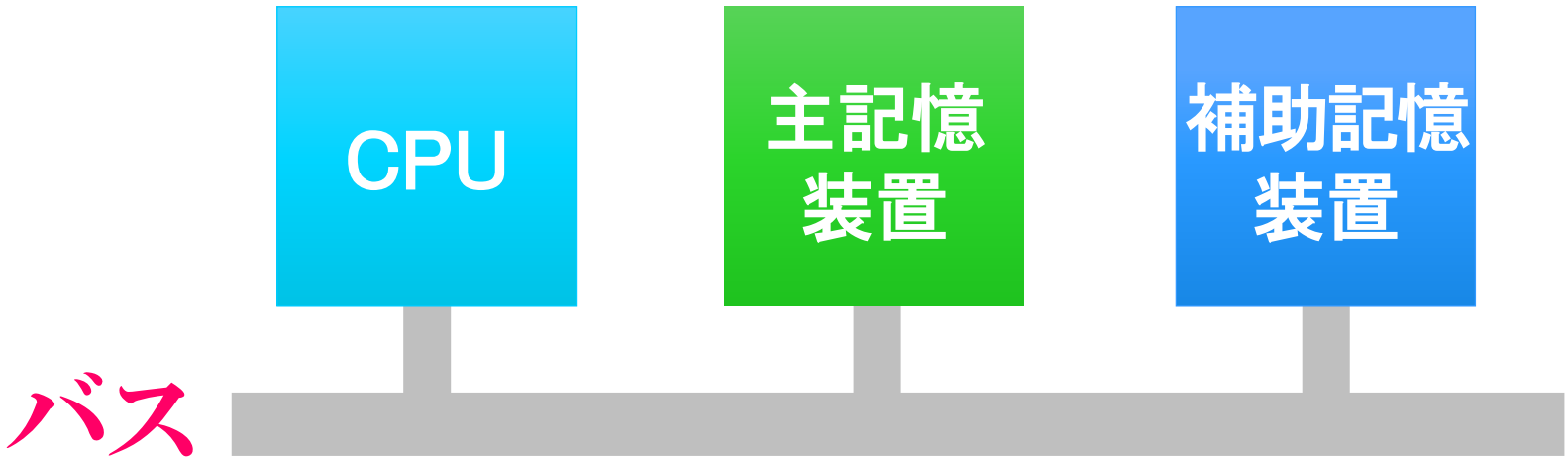
プログラムの実行


命令を1個ずつ順番に
読み出す
プログラムA
プログラムB
プログラムC
データA
データB
プログラムC
データA
命令
データ
補助記憶装置
(ハードディスク等)
主記憶装置
(メインメモリ)
CPU

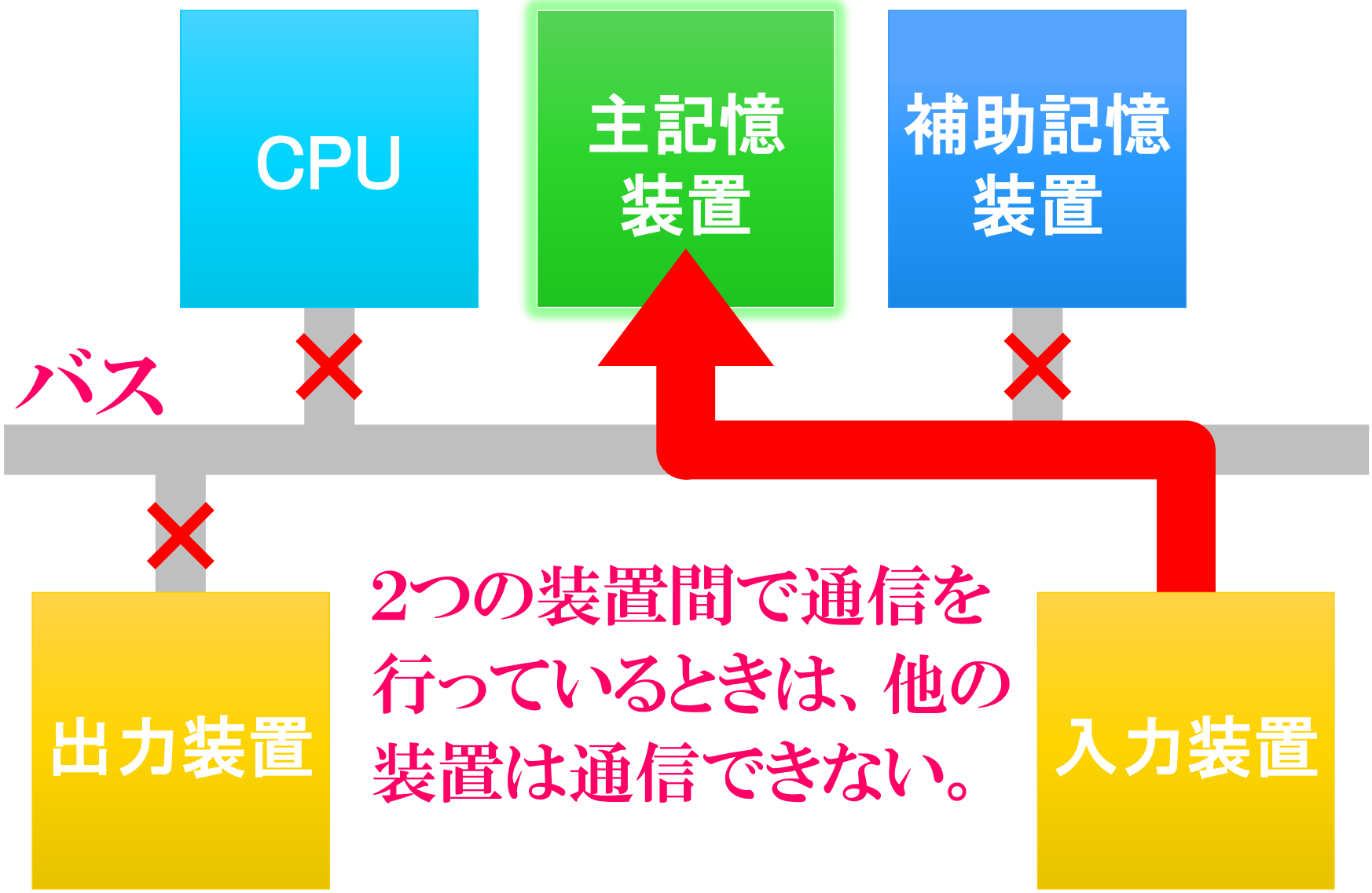
バス方式

バス

コンピュータ内の各装置がデータ通信に用いる共通の通信経路



バスによる装置間の通信



バスの種類

重要

- データバス

  データを送る。

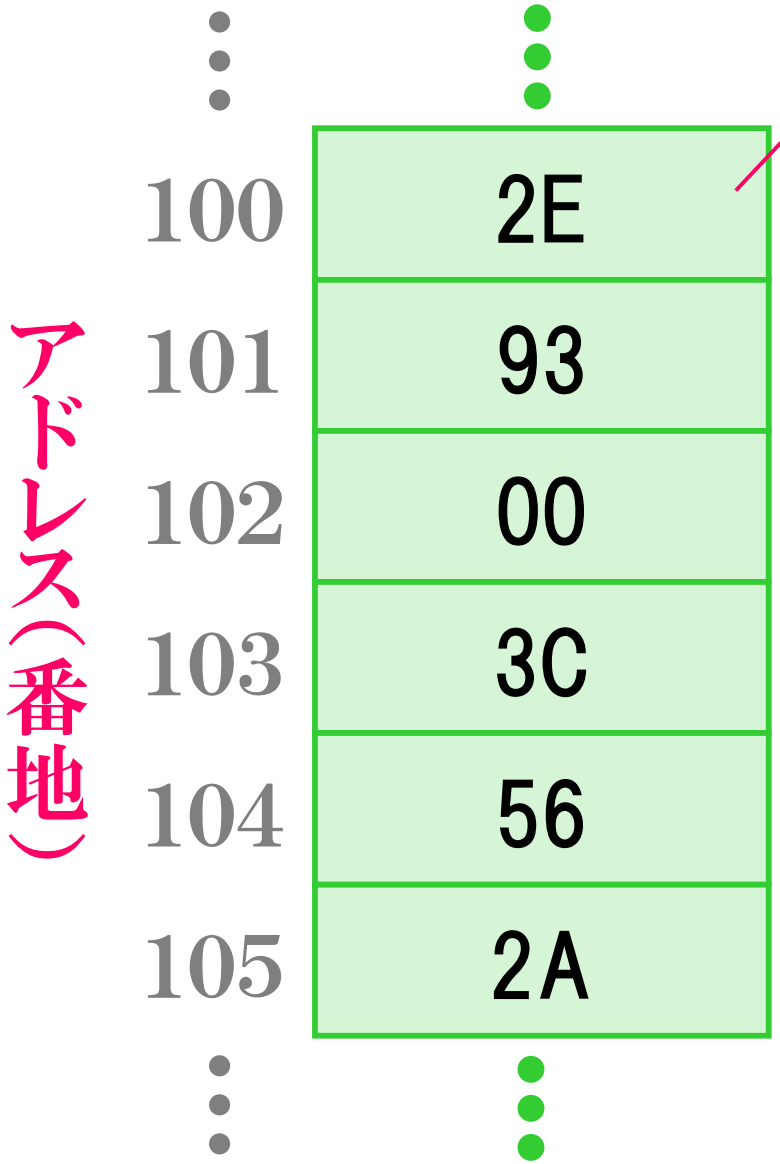
- アドレスバス

  主記憶装置のアドレス（番地）を送る。

- コントロールバス

  データの読み書きの制御信号を送る。

主記憶装置



記憶場所

数値化された命令やデータを記憶

主記憶装置内のデータを読み書きするときは、対象のデータのアドレスを指定する必要がある。

機械語とアセンブリ言語

CPUの機能	命令番号	命令語
データ転送	10	LD
加算	24	ADDA
減算	25	SUBA
論理積	34	AND
論理和	35	OR

機械語　アセンブリ言語

CPUの機能1つ1つに命令番号が与えられている。

CPUの構成要素

重要

✤レジスタ

略称	名称
MDR	記憶データ用レジスタ
MAR	記憶番地指定用レジスタ
PC	プログラムカウンタ
IR	命令レジスタ
GR0～GR7	汎用レジスタ
FR	フラグレジスタ

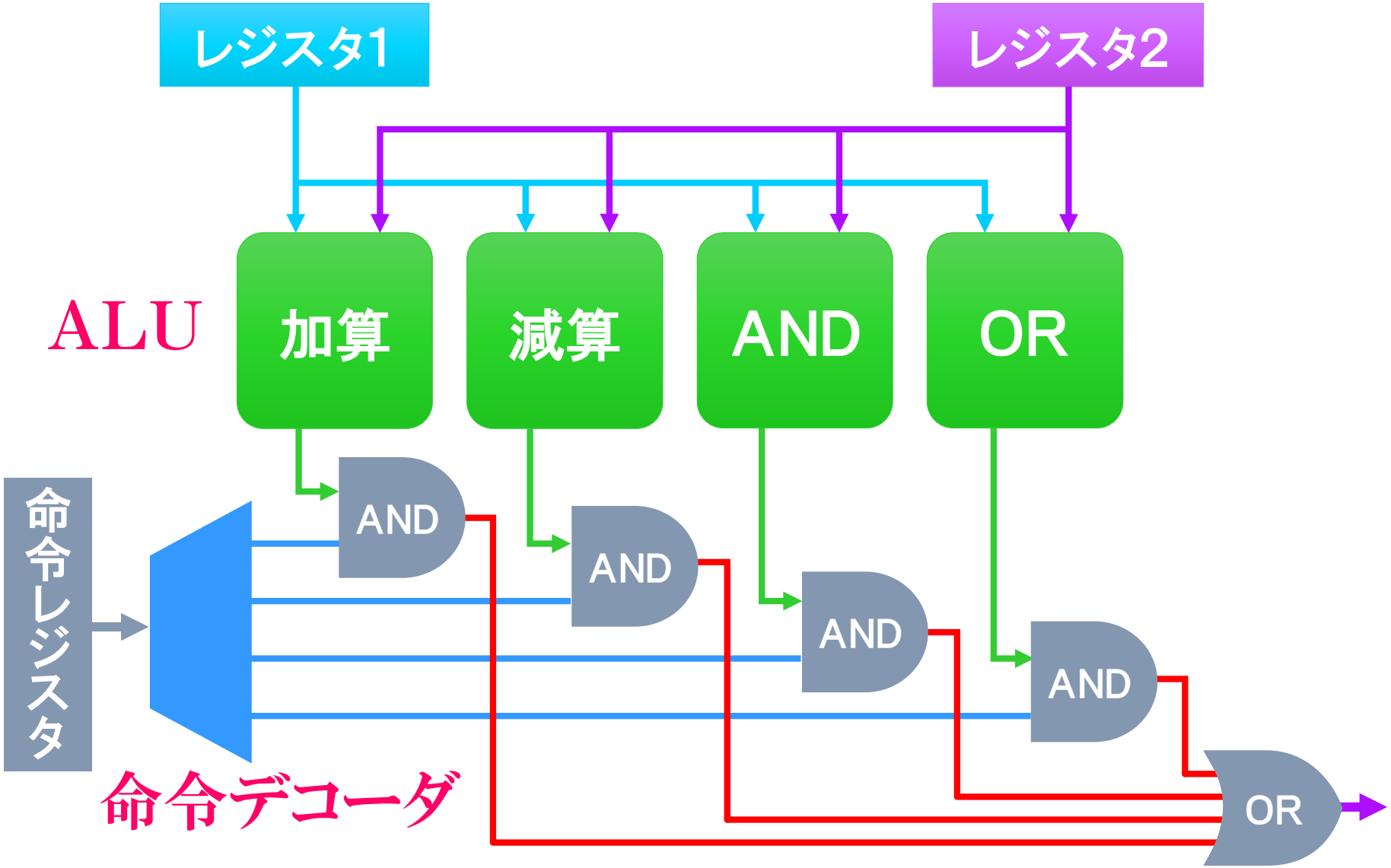
CPUの構成要素

重要

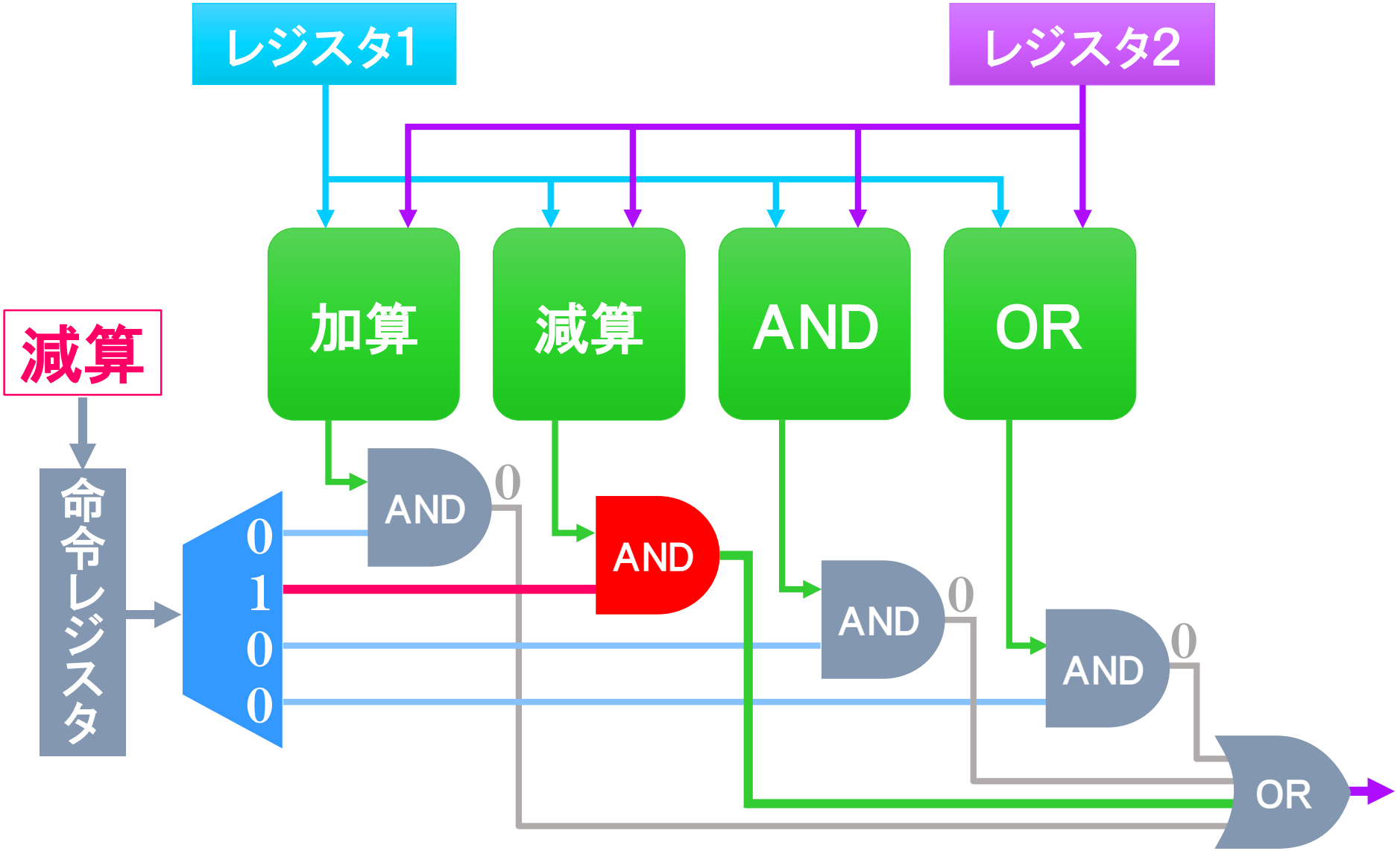
✤その他装置

略称・名称	働き
ALU	算術演算や論理演算を行う
命令デコーダ	命令番号を解読し、各装置を制御する
コントロールユニット	コントロールバスに制御信号を送る

命令デコーダとALU



減算を行う場合



命令サイクル

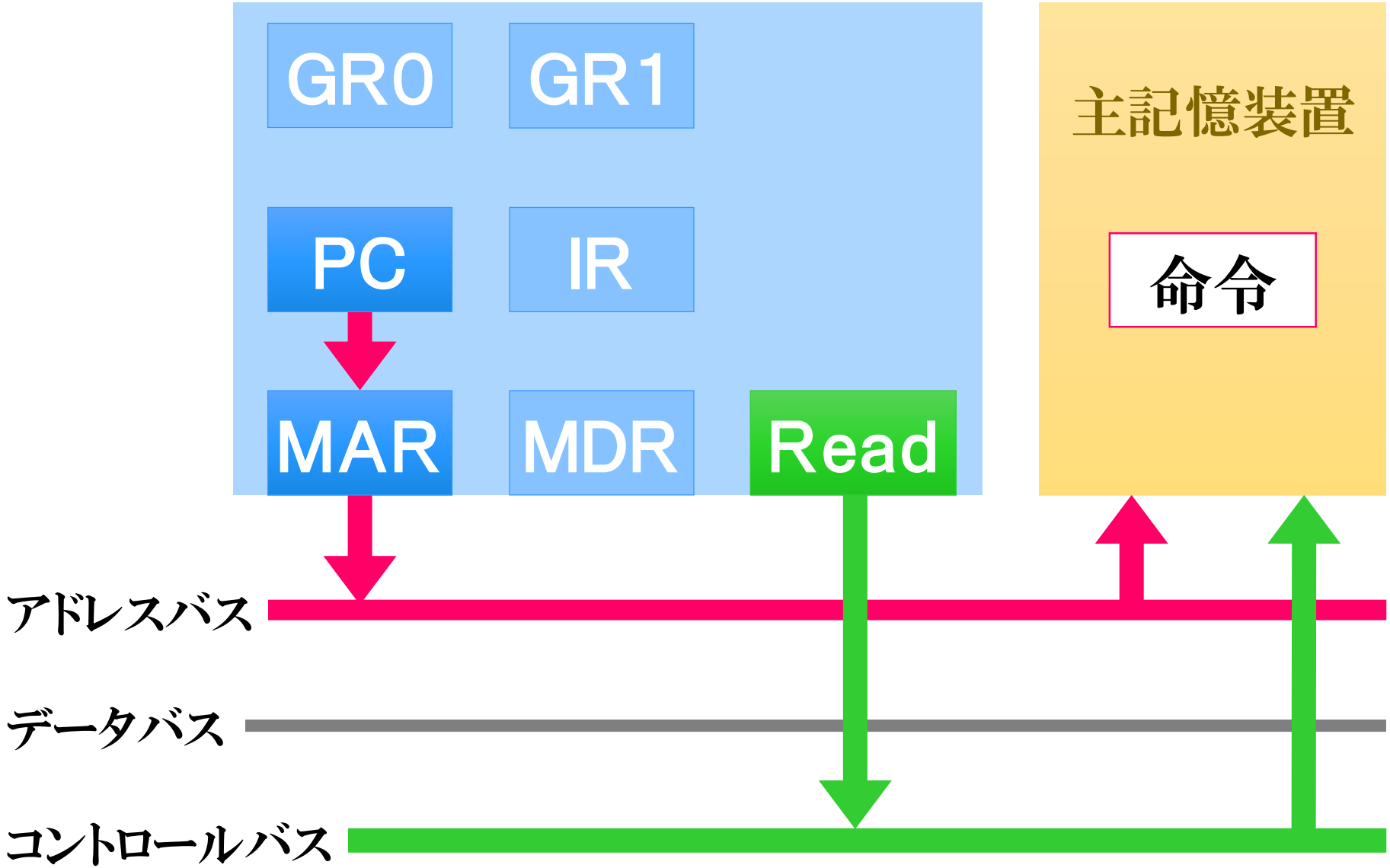
重要

1つの命令を実行する手順

1. 命令読み出し(fetch)
2. 命令解読(decode)
3. 命令実行(execute)
4. 結果書き込み(writeback)

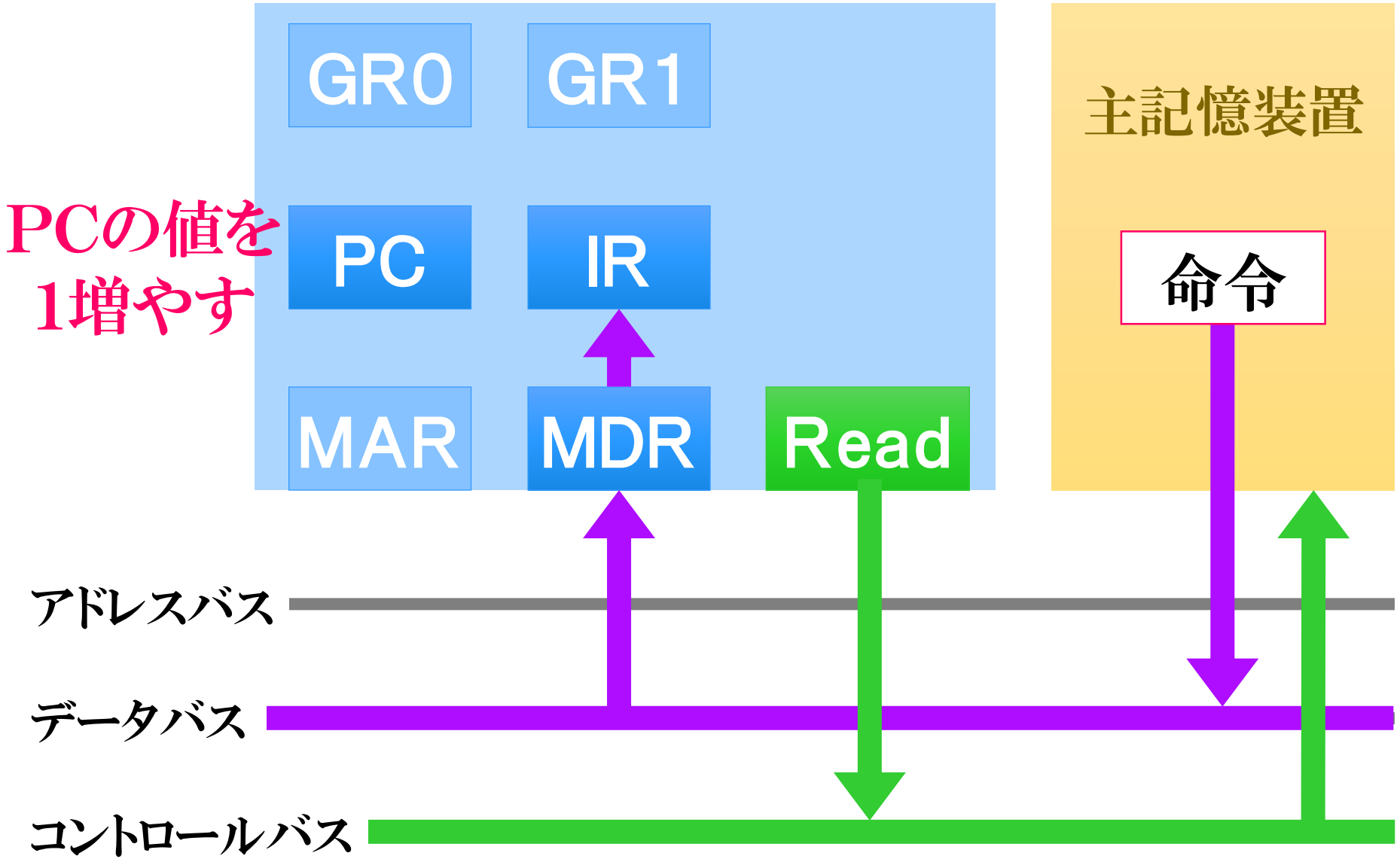
命令読み出し

重要



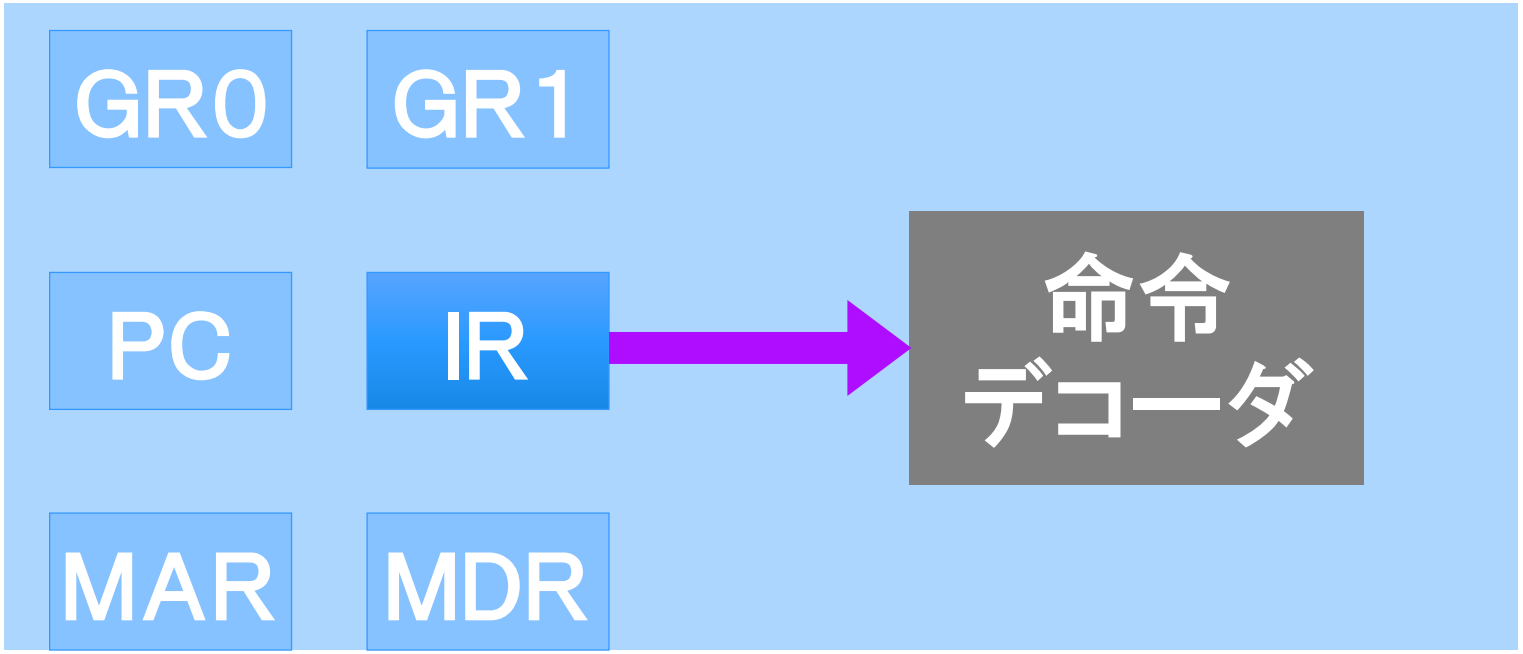
命令読み出し

重要


GR0
GR1
主記憶装置
PCの値を
1増やす
PC
IR
命令
MAR
MDR
Read
アドレスバス
データバス
コントロールバス

命令解読

重要



GR1←GR1＋GR0

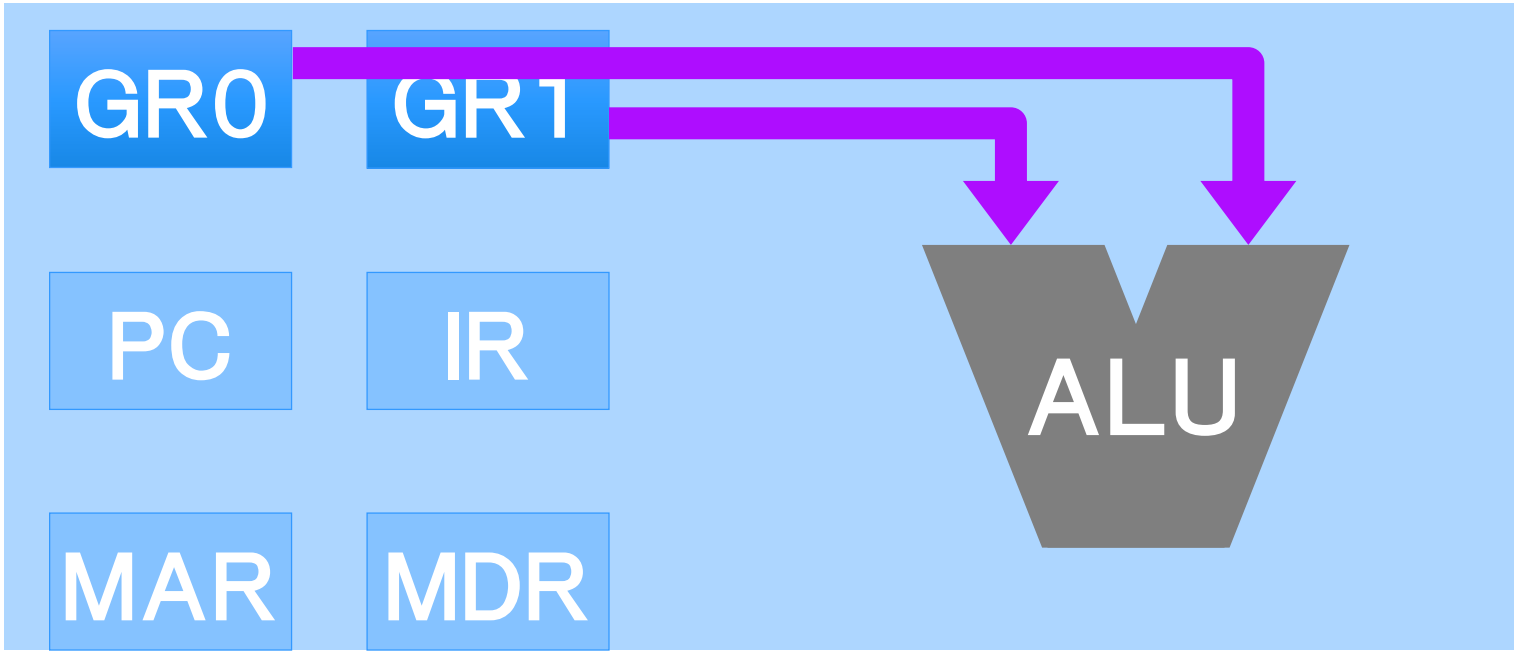
アドレスバス

データバス

コントロールバス

命令実行

重要



GR1←GR1＋GR0

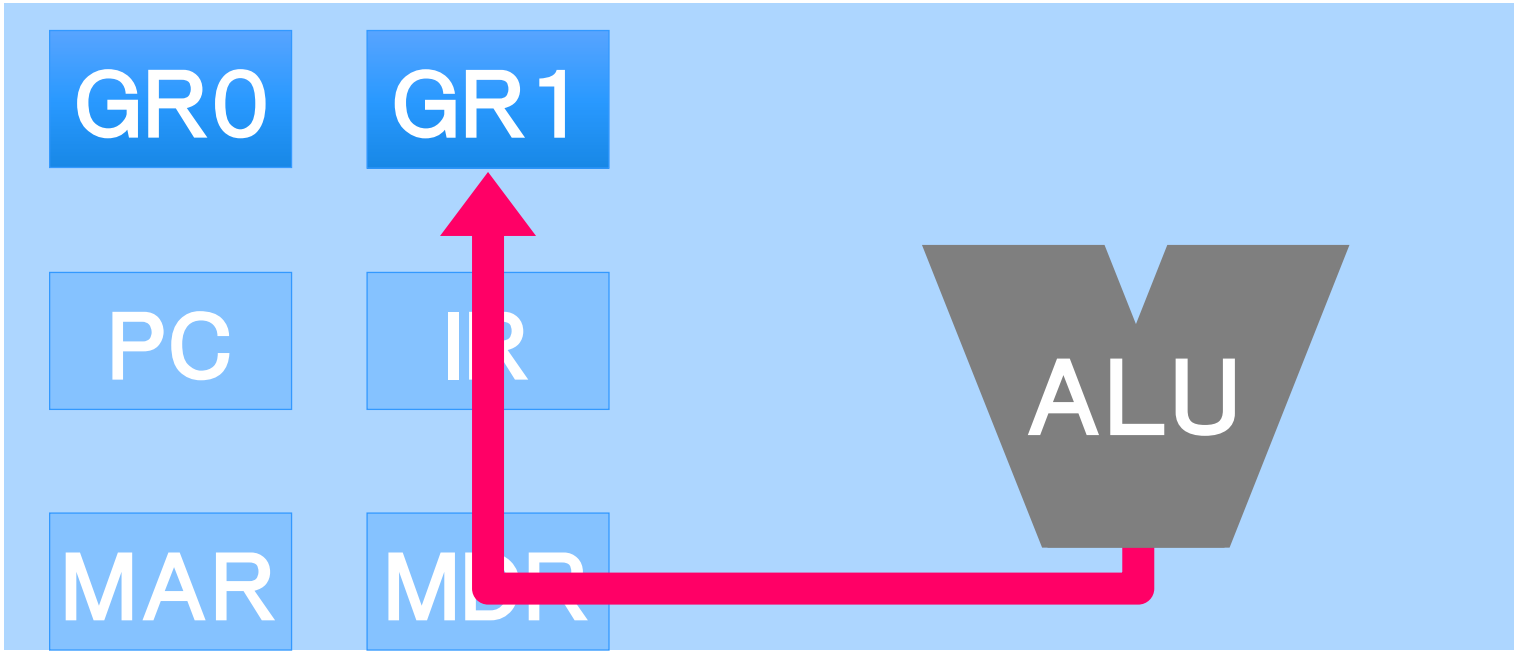
アドレスバス

データバス

コントロールバス

結果書き込み

重要



GR1←GR1＋GR0

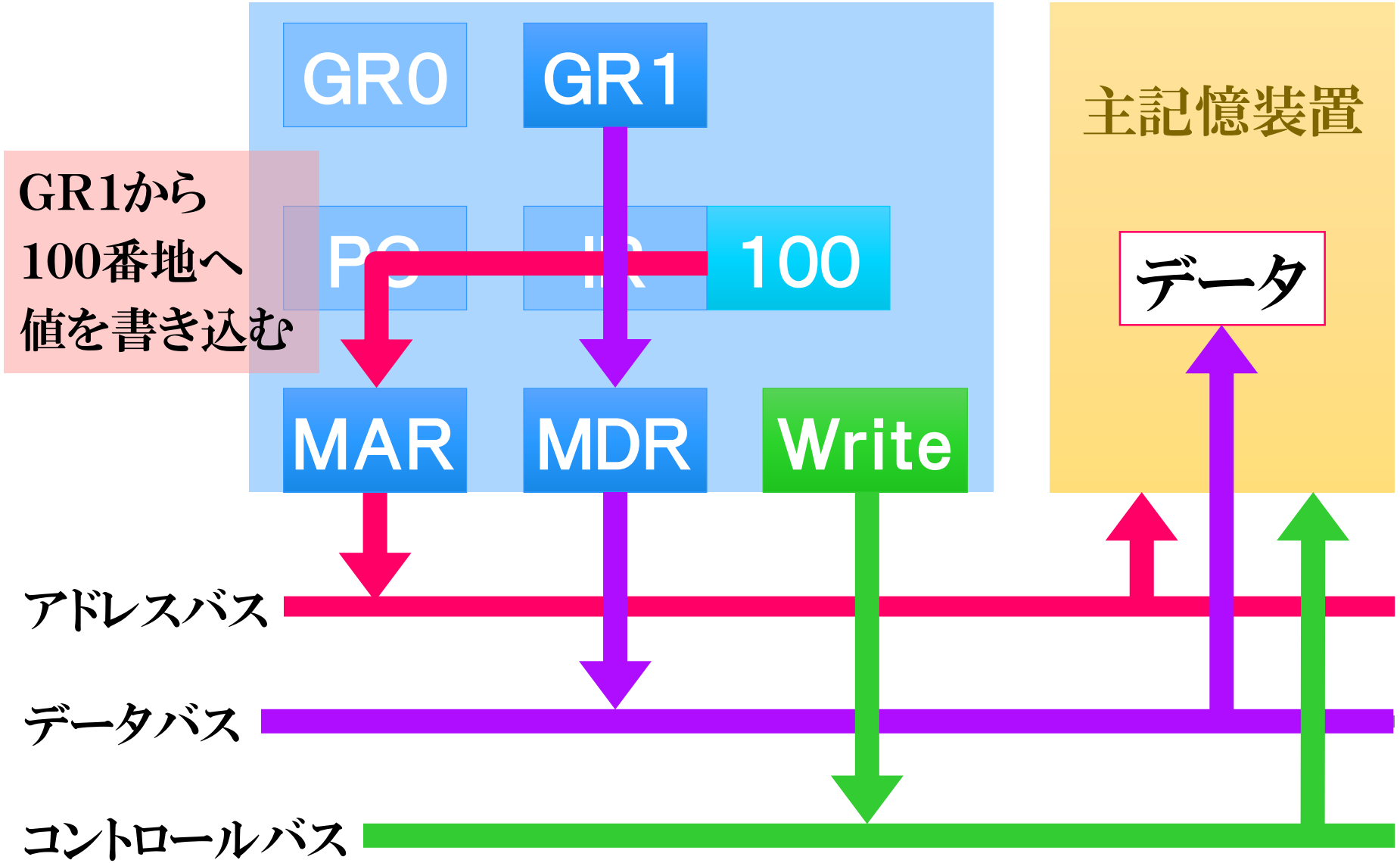
アドレスバス

データバス

コントロールバス

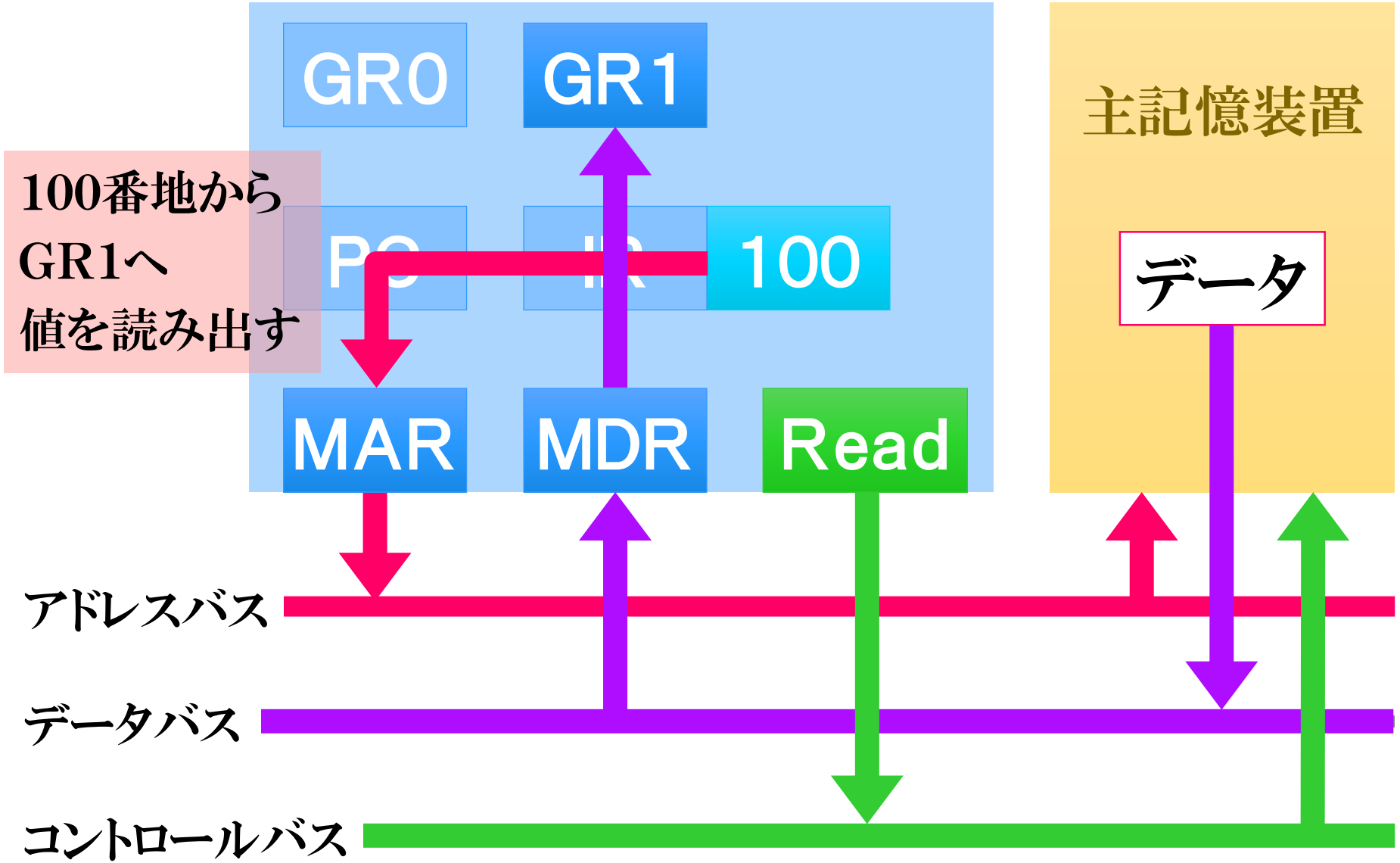
主記憶装置への書き込み

重要



主記憶装置からの読み出し

重要

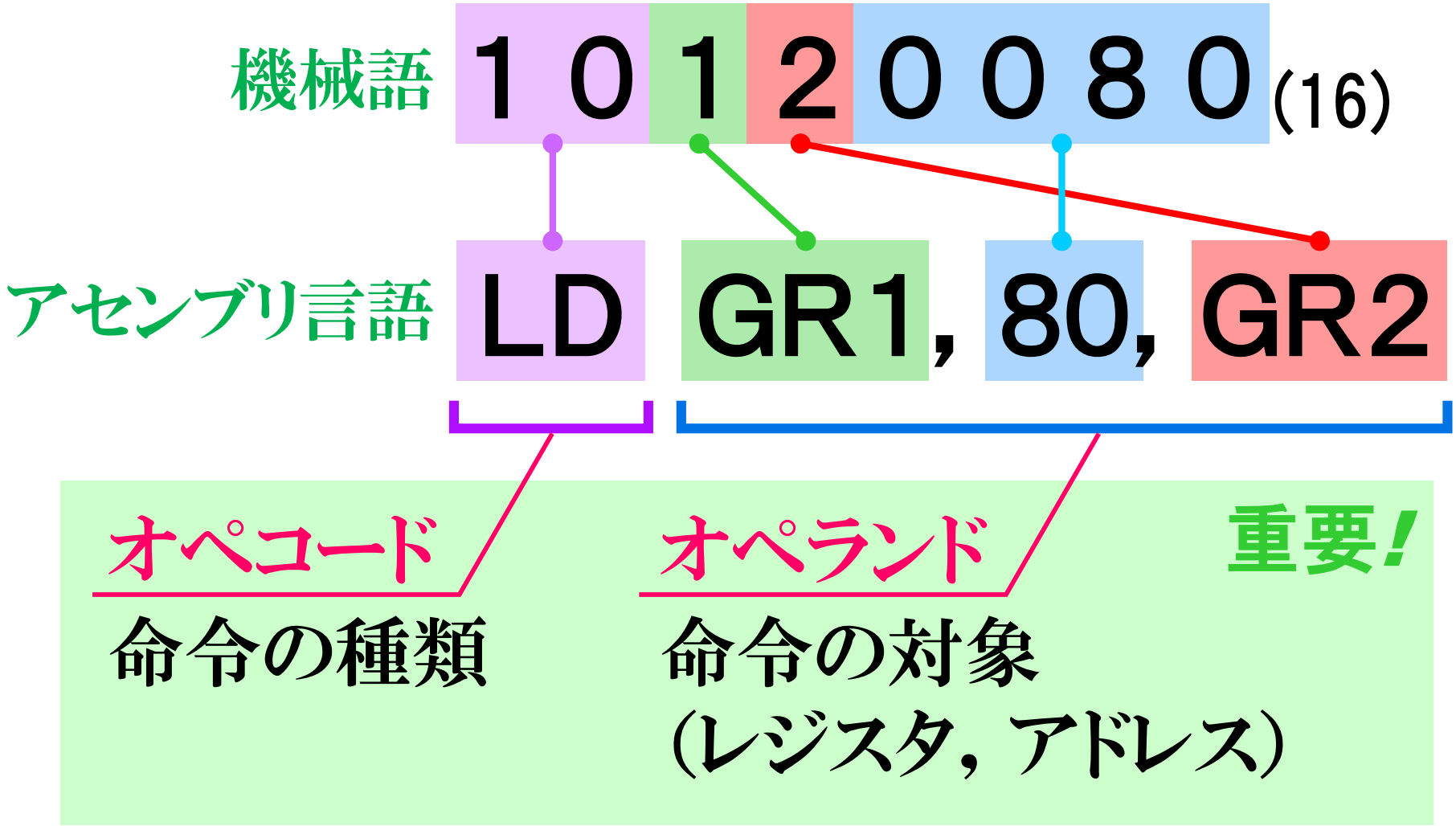


アセンブリ言語

CASLⅡを取り扱う。

- データ転送命令
- 算術演算命令(加算、減算)
- 分岐命令
- 算術比較命令

命令の記述


機械語
1 0 1 2 0 0 8 0 (16)
アセンブリ言語
LD GR1, 80, GR2
オペコード
命令の種類
オペランド
命令の対象
(レジスタ, アドレス)
重要!

LD（Load）

重要

レジスタ、または、主記憶装置内の値を、指定したレジスタに読み出す。

LD　r1，r2

レジスタr1 ← レジスタr2の値

LD　r1，adr

レジスタr1 ← adr番地の値

ST（Store）

重要

レジスタの値を、主記憶装置内の指定したアドレスに書き込む。

ST　r1，adr

adr番地 ← レジスタr1の値

LAD（Load Address）

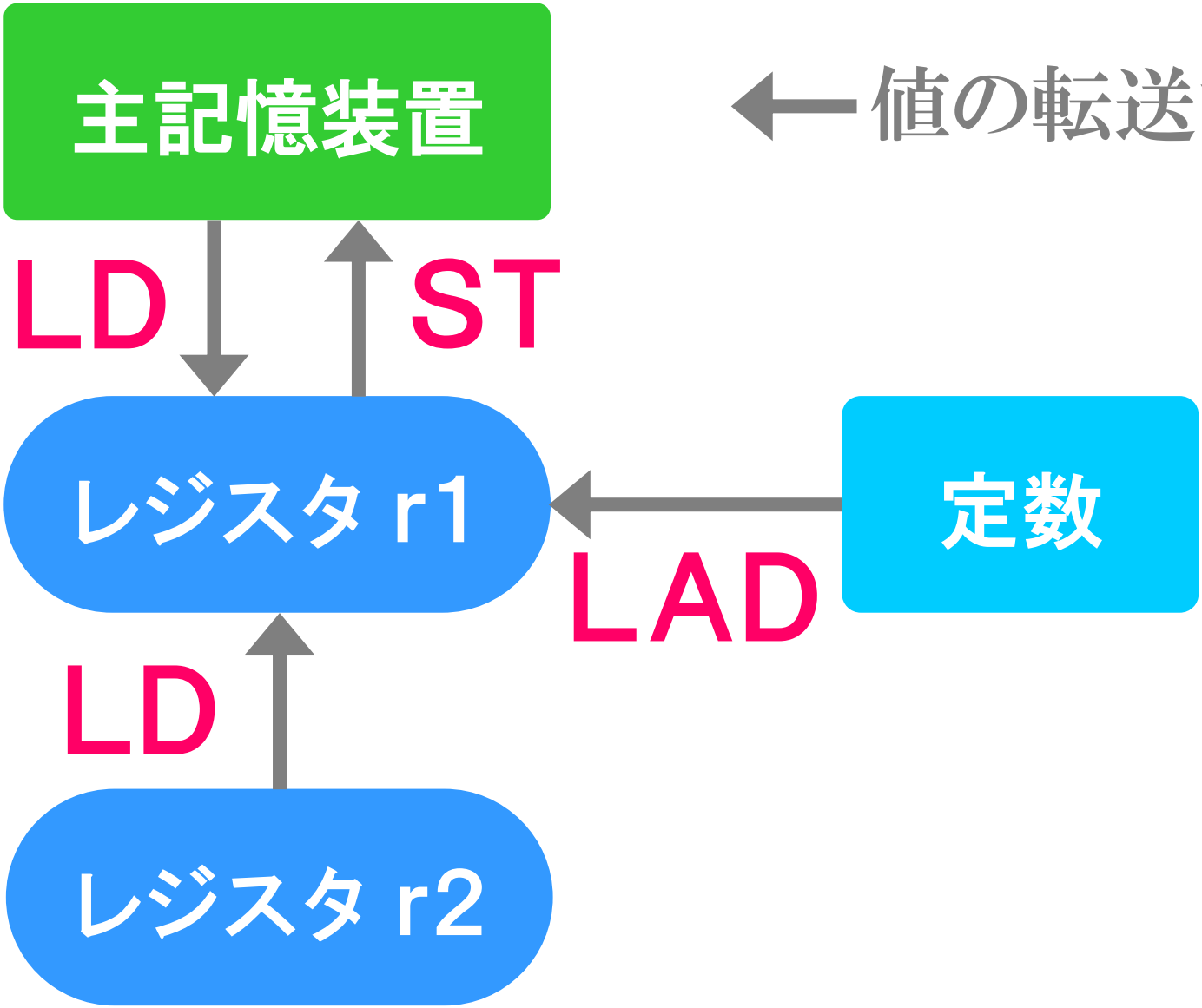
重要

レジスタに定数（番地）を書き込む。

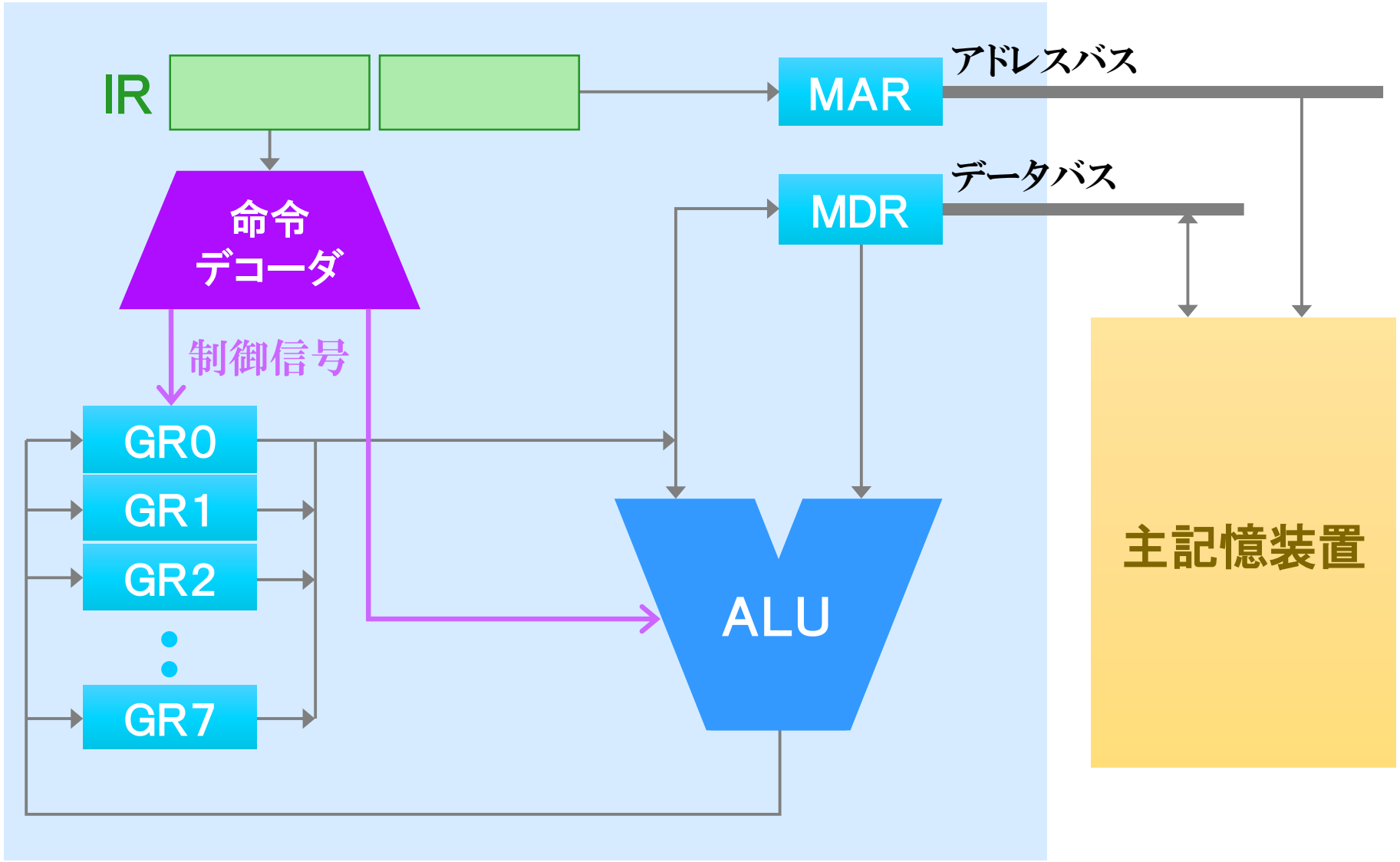
LAD　r1，adr

レジスタr1 ← adr

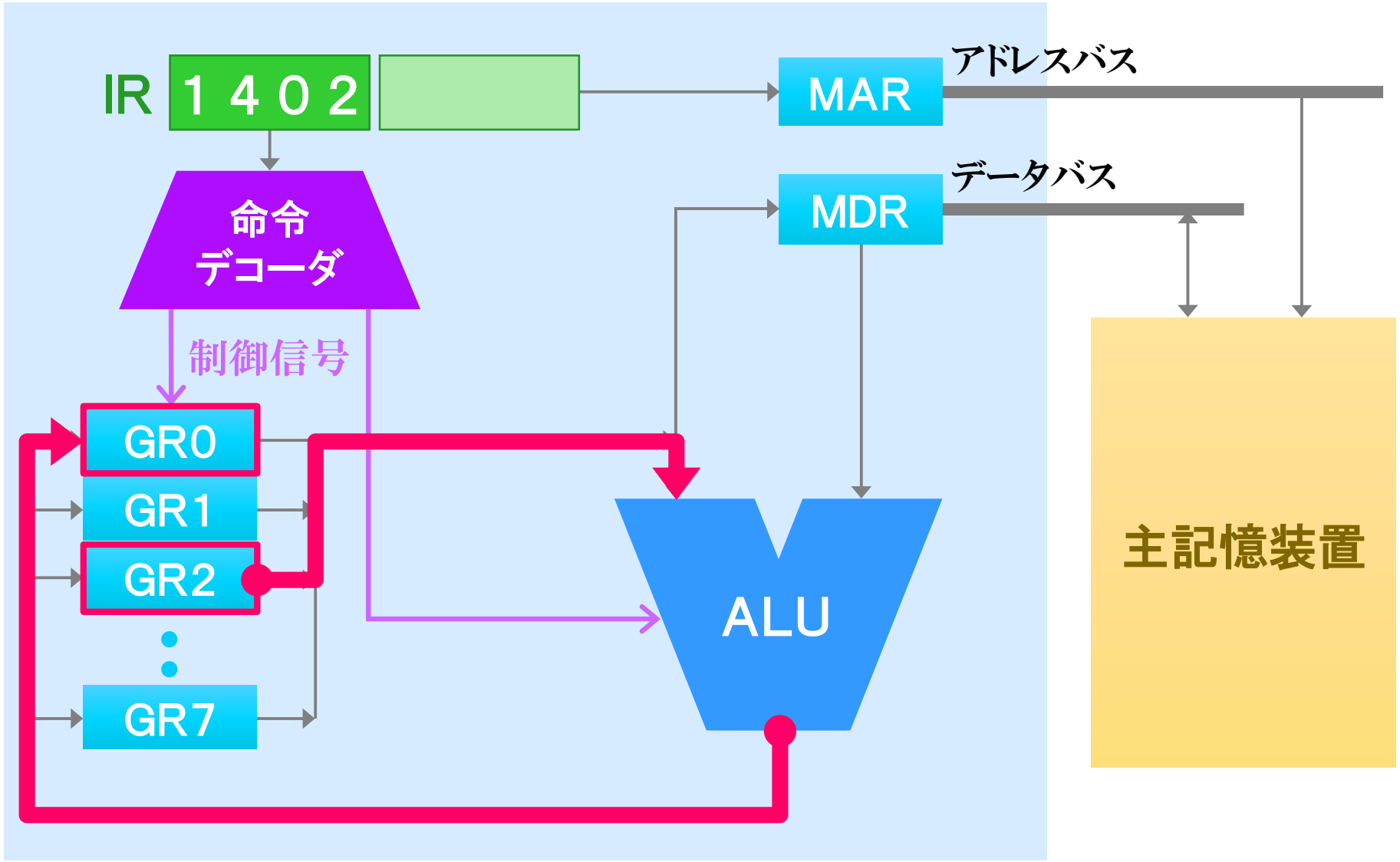
データ転送命令のまとめ



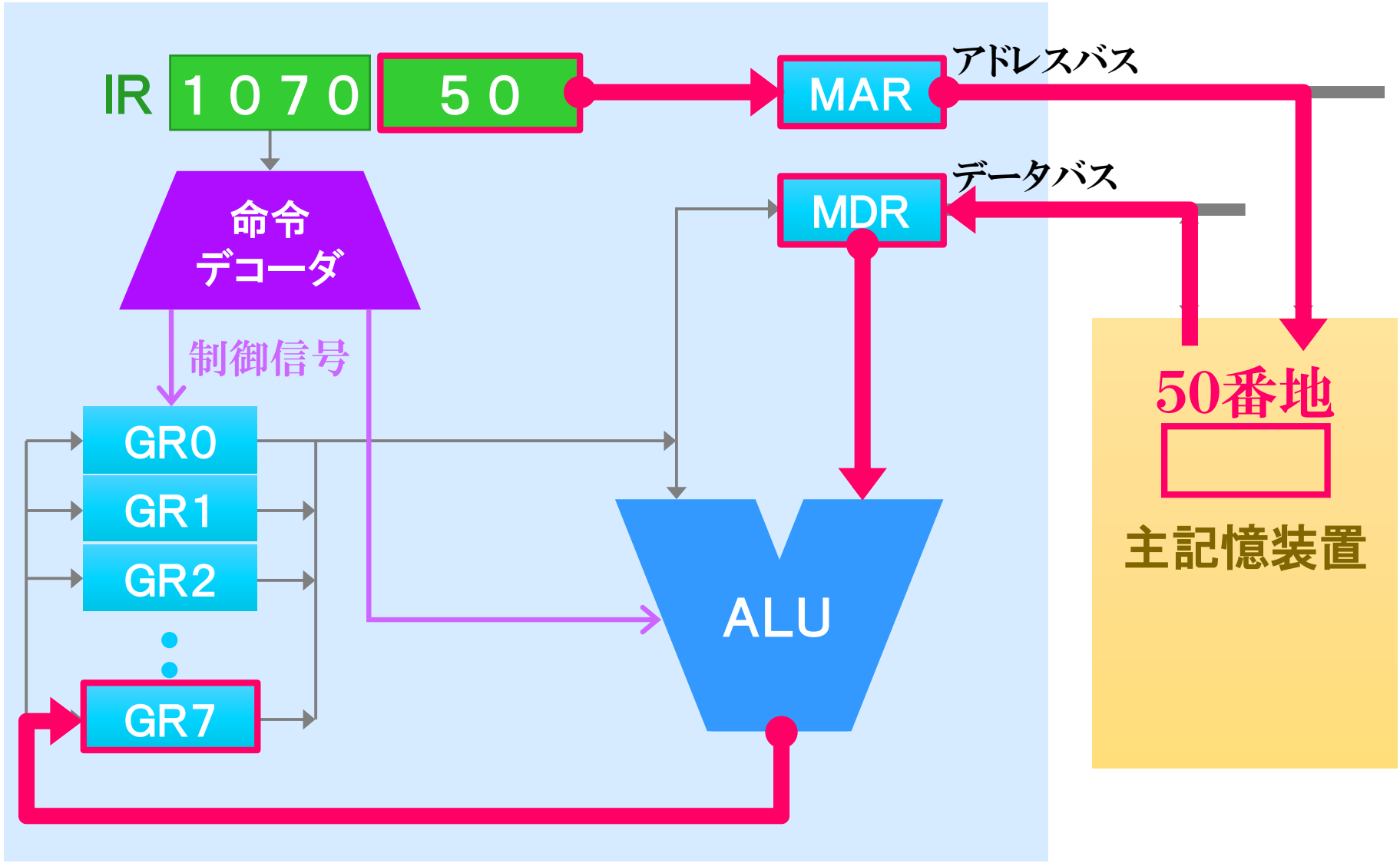
CPUの概略図



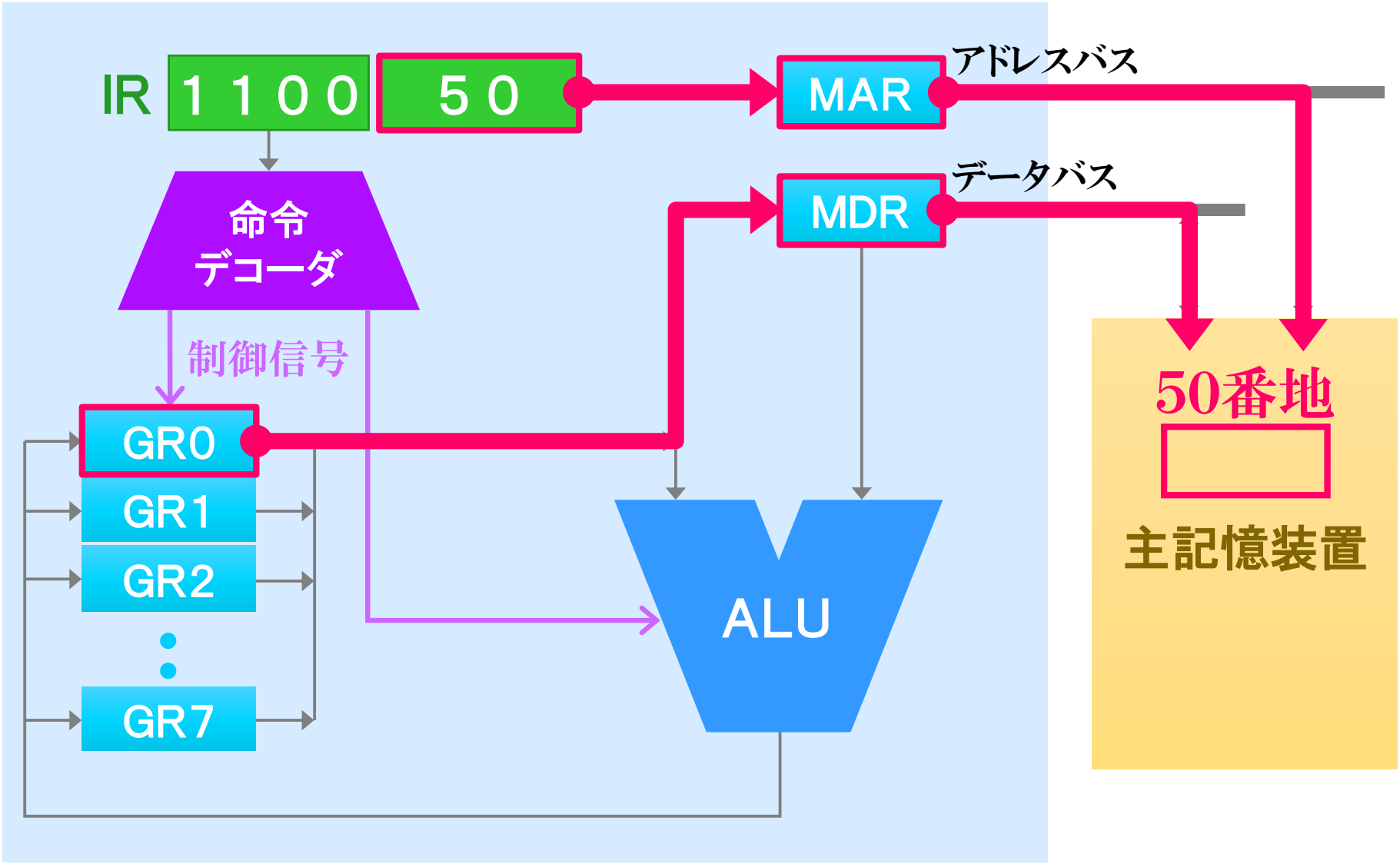
LD GR0, GR2


IR
1 4 0 2
MAR
アドレスバス
MDR
データバス
命令
デコーダ
制御信号
GR0
GR1
GR2
GR7
ALU
主記憶装置

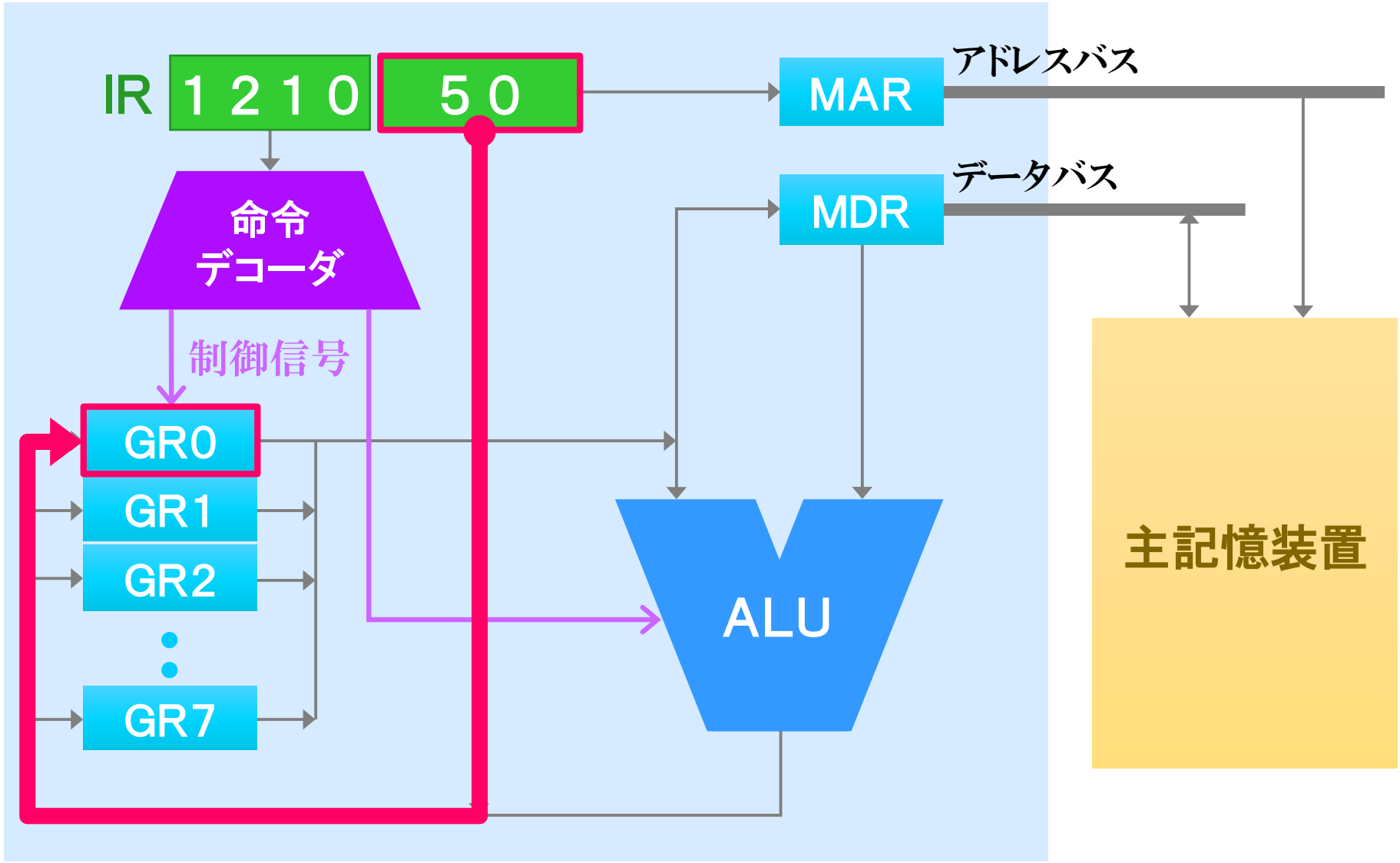
LD GR7, 50


IR
1 0 7 0
5 0
MAR
アドレスバス
MDR
データバス
命令
デコーダ
制御信号
GR0
GR1
GR2
GR7
ALU
50番地
主記憶装置

ST GR0, 50


IR
1 1 0 0
5 0
MAR
アドレスバス
MDR
データバス
命令
デコーダ
制御信号
GR0
GR1
GR2
GR7
ALU
50番地
主記憶装置

LAD GR1, 50


IR
1 2 1 0
5 0
命令
デコーダ
制御信号
GR0
GR1
GR2
GR7
MAR
アドレスバス
MDR
データバス
ALU
主記憶装置

ADDA（Add Arithmetic）

2つの値の加算結果をレジスタに書き込む。

重要

ADDA　r1，r2

レジスタr1 ← r1の値 ＋ r2の値

ADDA　r1，adr

レジスタr1 ← r1の値 ＋ adr番地の値

SUBA（Subtract Arithmetic）

2つの値の減算結果をレジスタに書き込む。

重要

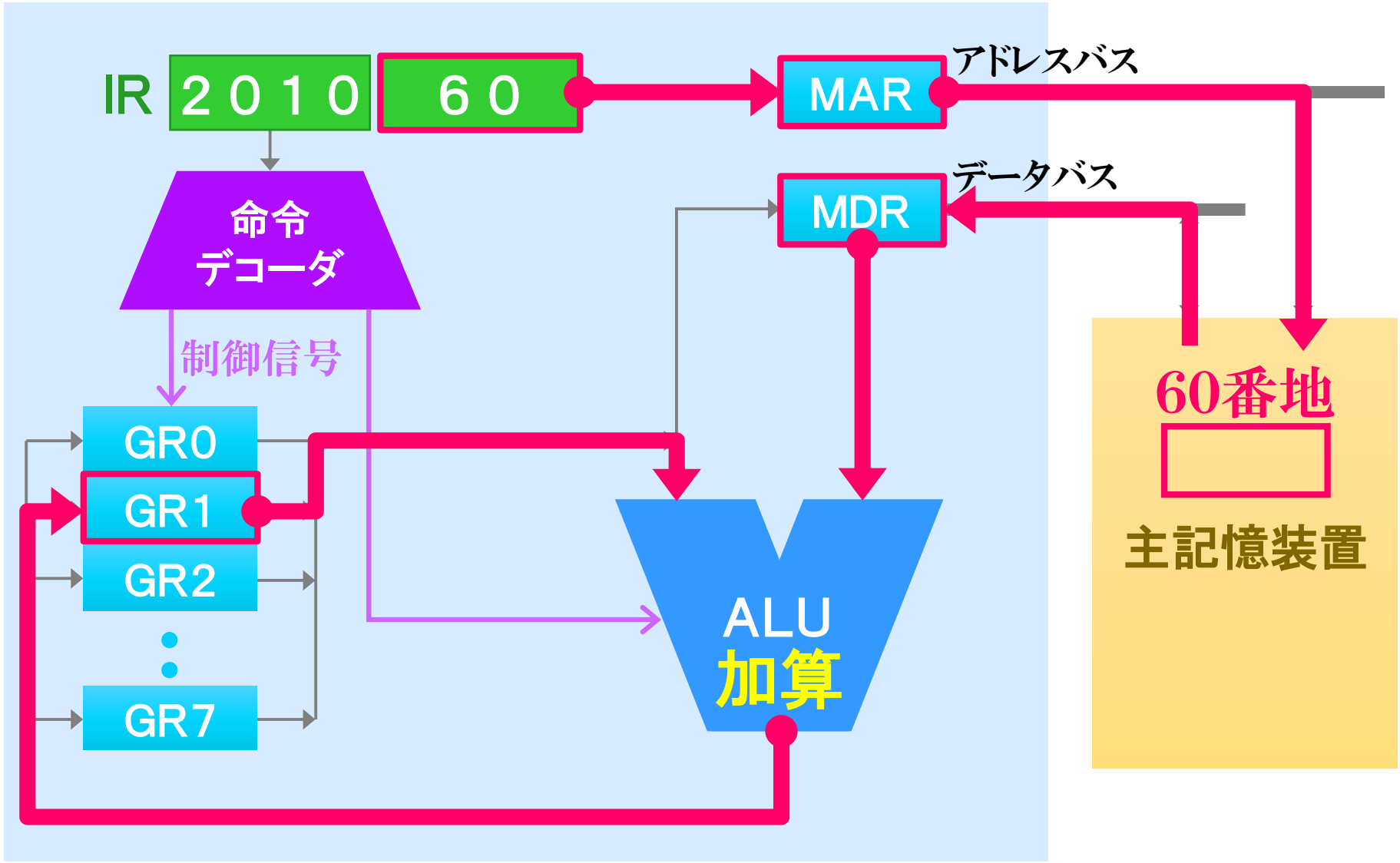
SUBA r1, r2

レジスタr1 ← r1の値 － r2の値

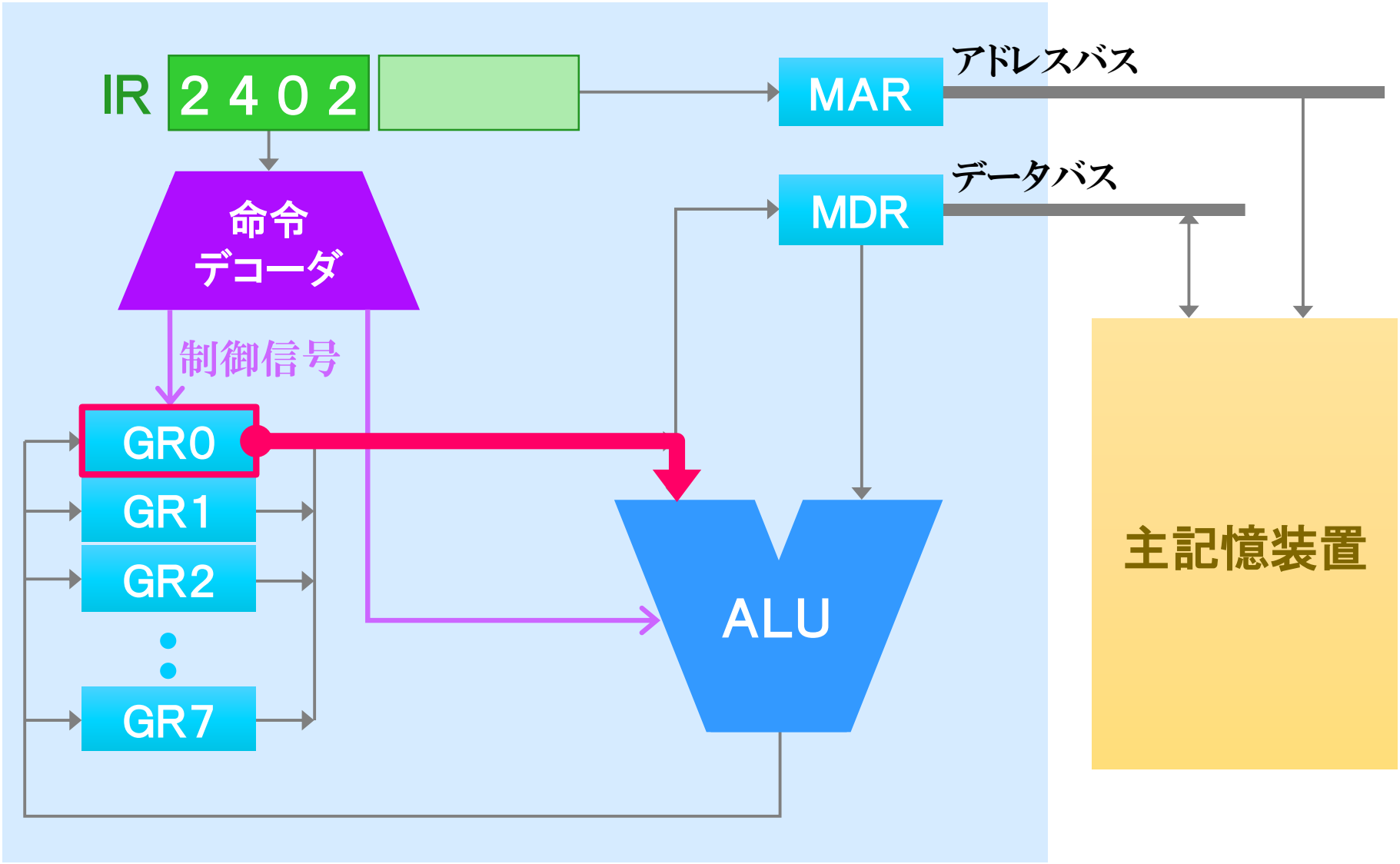
SUBA r1, adr

レジスタr1 ← r1の値 － adr番地の値

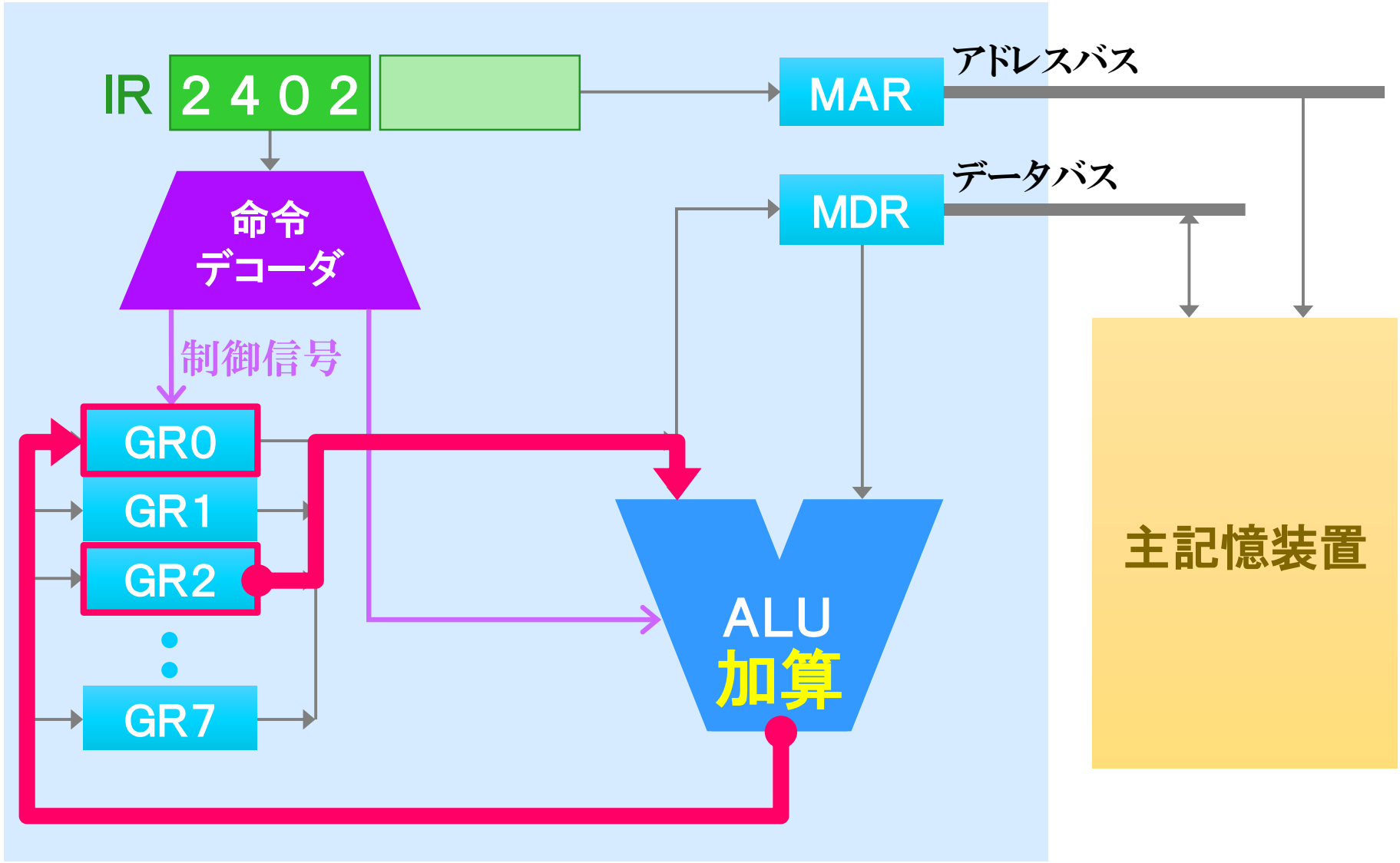
ADDA GR1, 60


IR
2010
60
MAR
アドレスバス
データバス
MDR
命令
デコーダ
制御信号
GR0
GR1
GR2
GR7
ALU
加算
60番地
主記憶装置

ADDA GR0, GR2


IR
2 4 0 2
MAR
アドレスバス
データバス
MDR
命令
デコーダ
制御信号
GR0
GR1
GR2
GR7
ALU
主記憶装置

ADDA GR0, GR2


IR
2 4 0 2
MAR
アドレスバス
MDR
データバス
命令
デコーダ
制御信号
GR0
GR1
GR2
GR7
ALU
加算
主記憶装置

実効アドレスと指標レジスタ 重要

LD r1，adr

レジスタr1 ← adr番地の値

実効アドレス

処理対象のアドレス

LD r1，adr，x

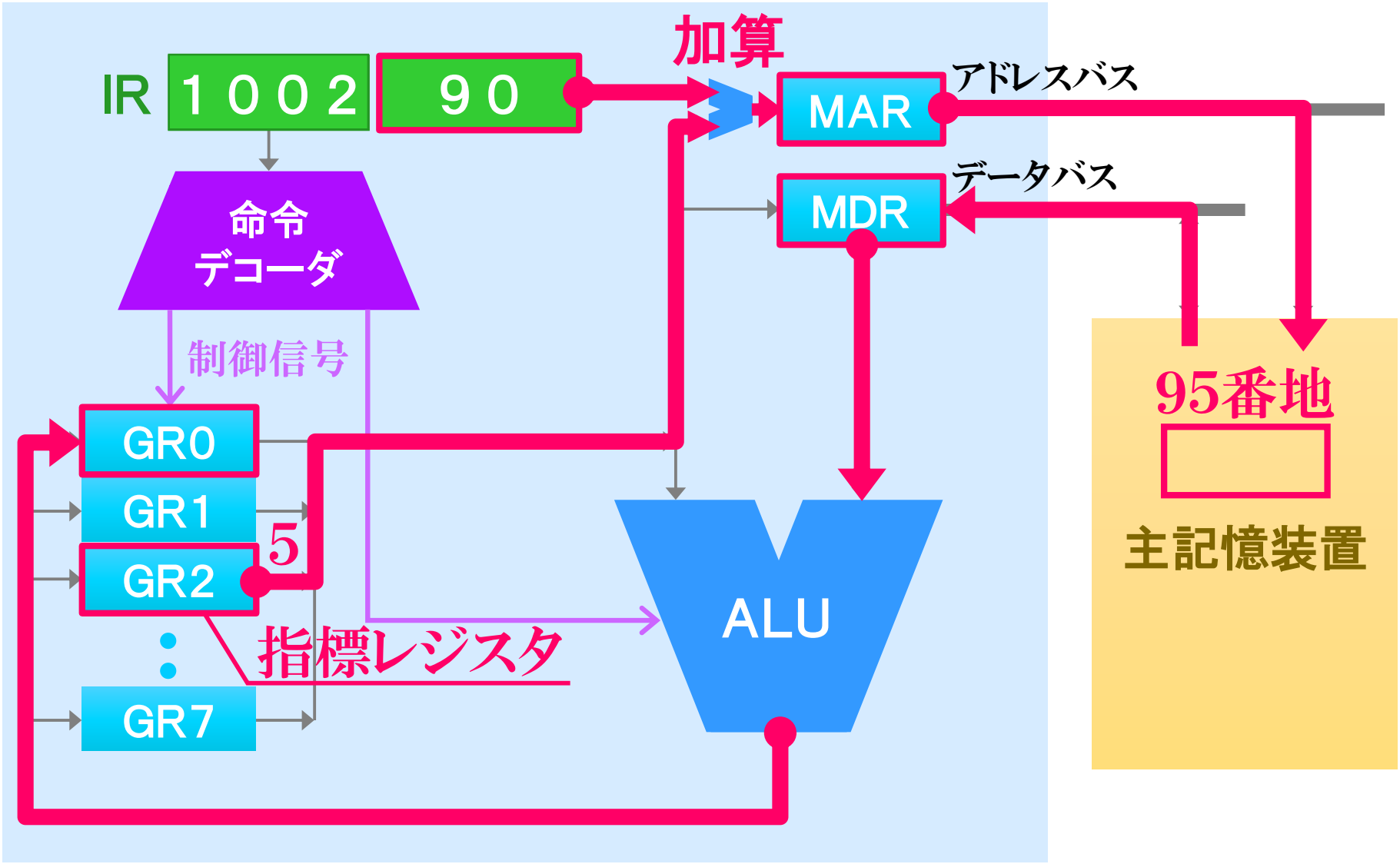
指標レジスタ

汎用レジスタの中の1つ

レジスタr1 ← (adr＋レジスタx) 番地の値

実効アドレス

LD GR0, 90, GR2


加算
IR
1002
90
MAR
アドレスバス
MDR
データバス
命令
デコーダ
制御信号
GR0
GR1
GR2
5
GR7
指標レジスタ
ALU
95番地
主記憶装置

指標レジスタを用いた命令記述

重要

LD　r1，adr，x

レジスタr1 ← (adr ＋ レジスタx) 番地の値

LAD　r1，adr，x

レジスタr1 ← (adr ＋ レジスタx)

ST　r1，adr，x

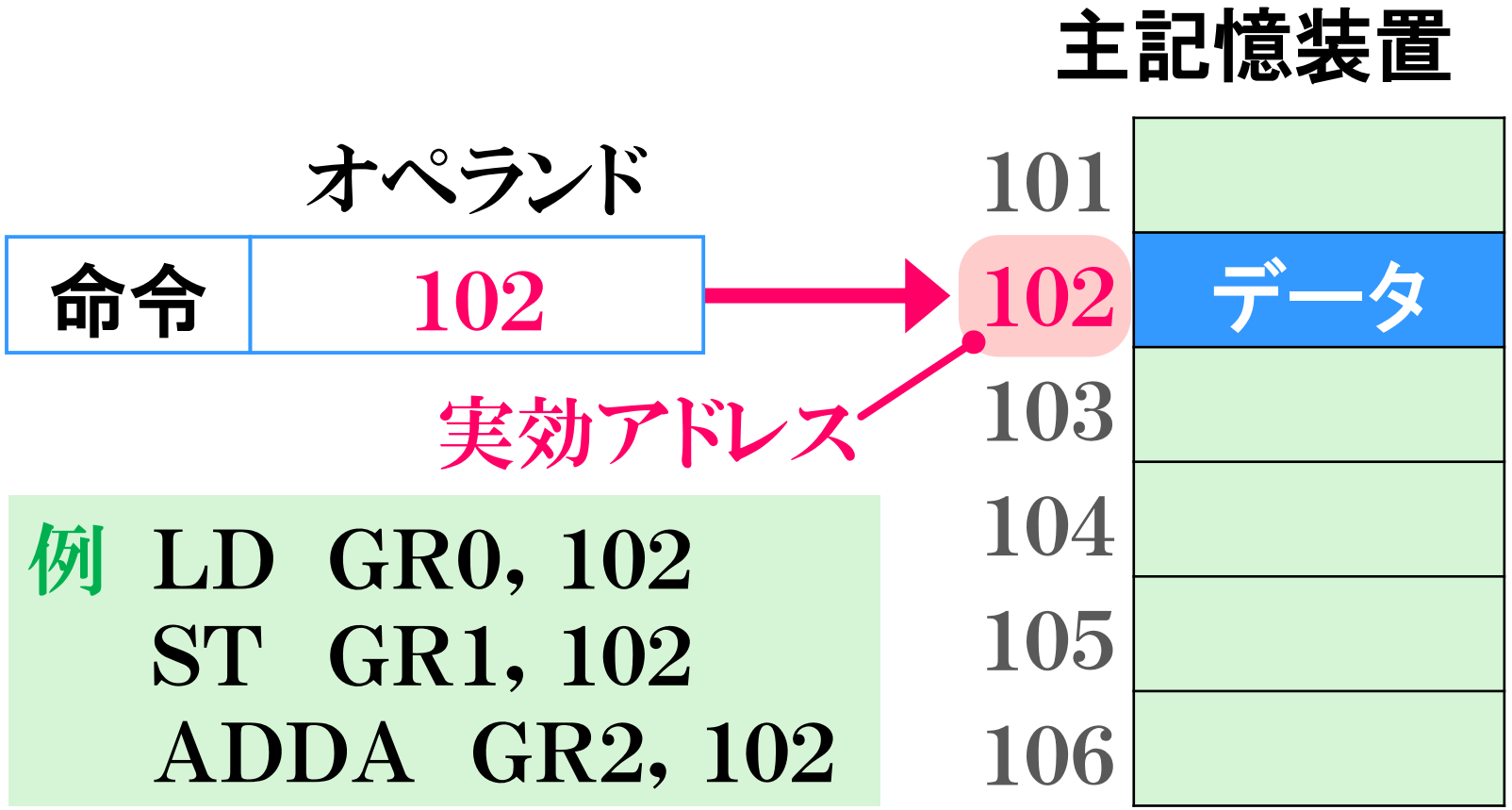
(adr ＋ レジスタx) 番地 ← レジスタr1の値

アドレス指定方式

- 直接アドレス指定
- 間接アドレス指定
- 相対アドレス指定
- 指標アドレス指定
- 即値アドレス指定

直接アドレス指定

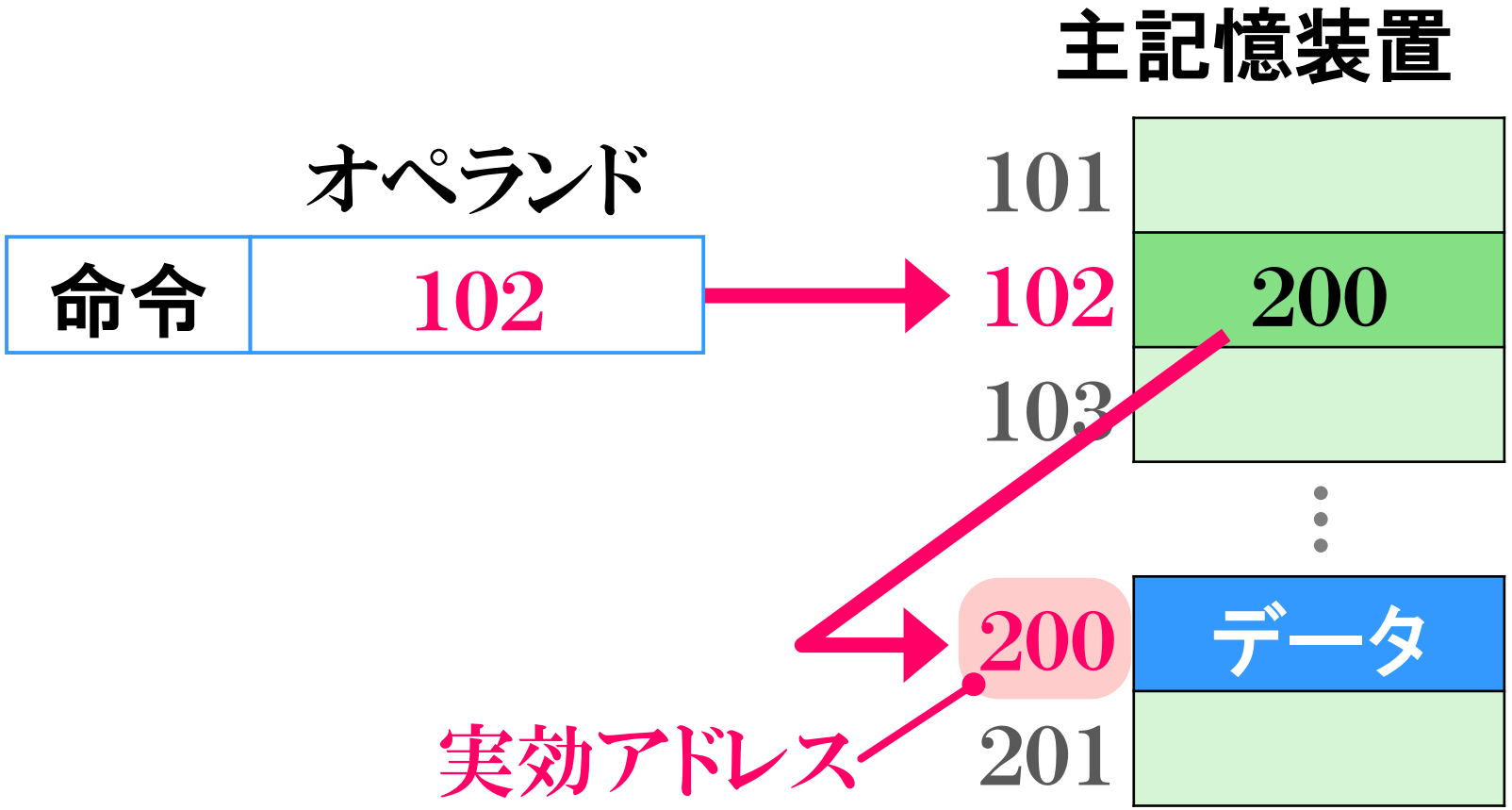
オペランドのアドレス部の値を実効アドレスとする方式



例
```
LD   GR0, 102
ST   GR1, 102
ADDA GR2, 102
```

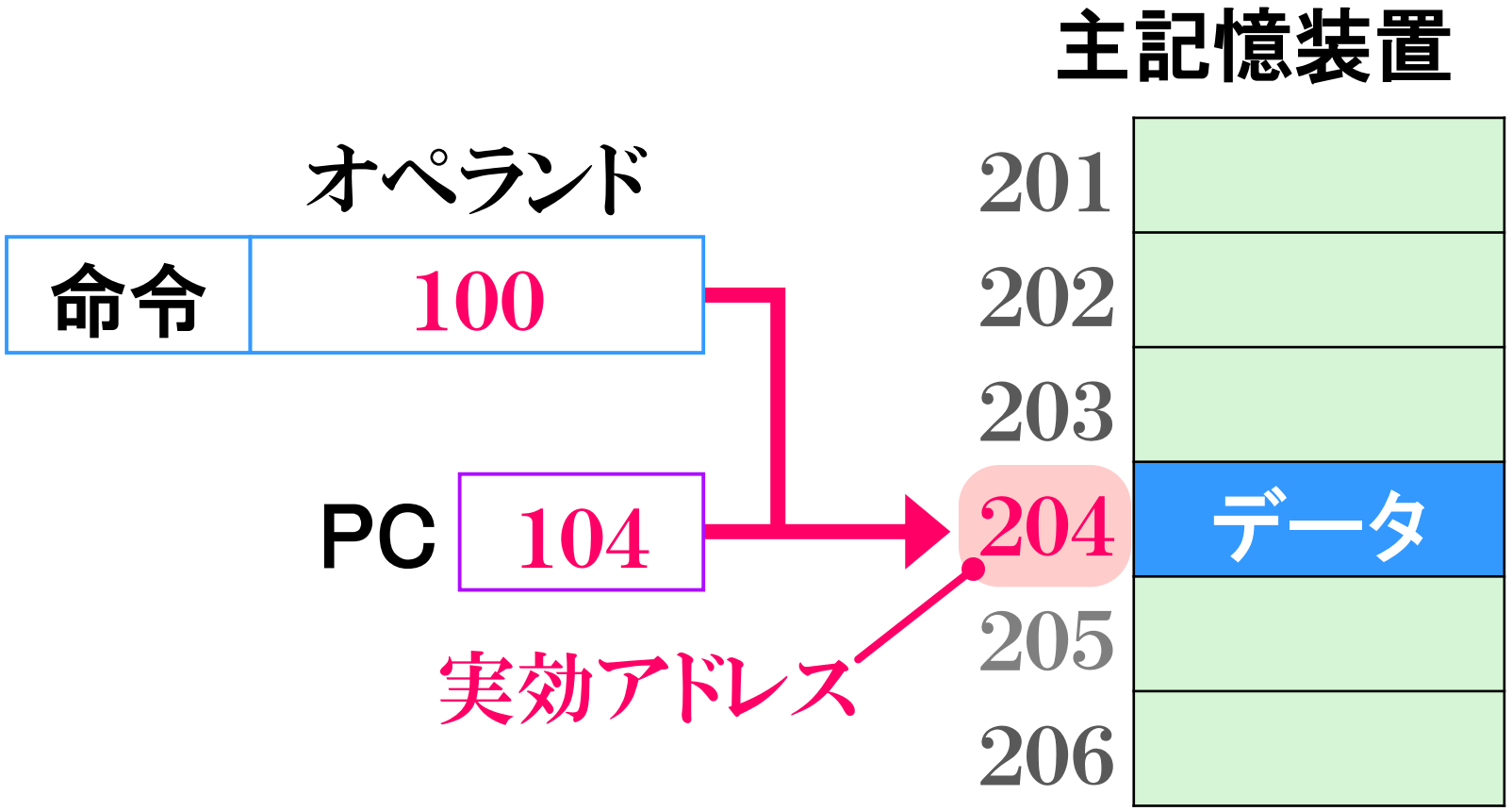
間接アドレス指定

オペランドのアドレス部に、実効アドレスを格納しているアドレスを格納する方式



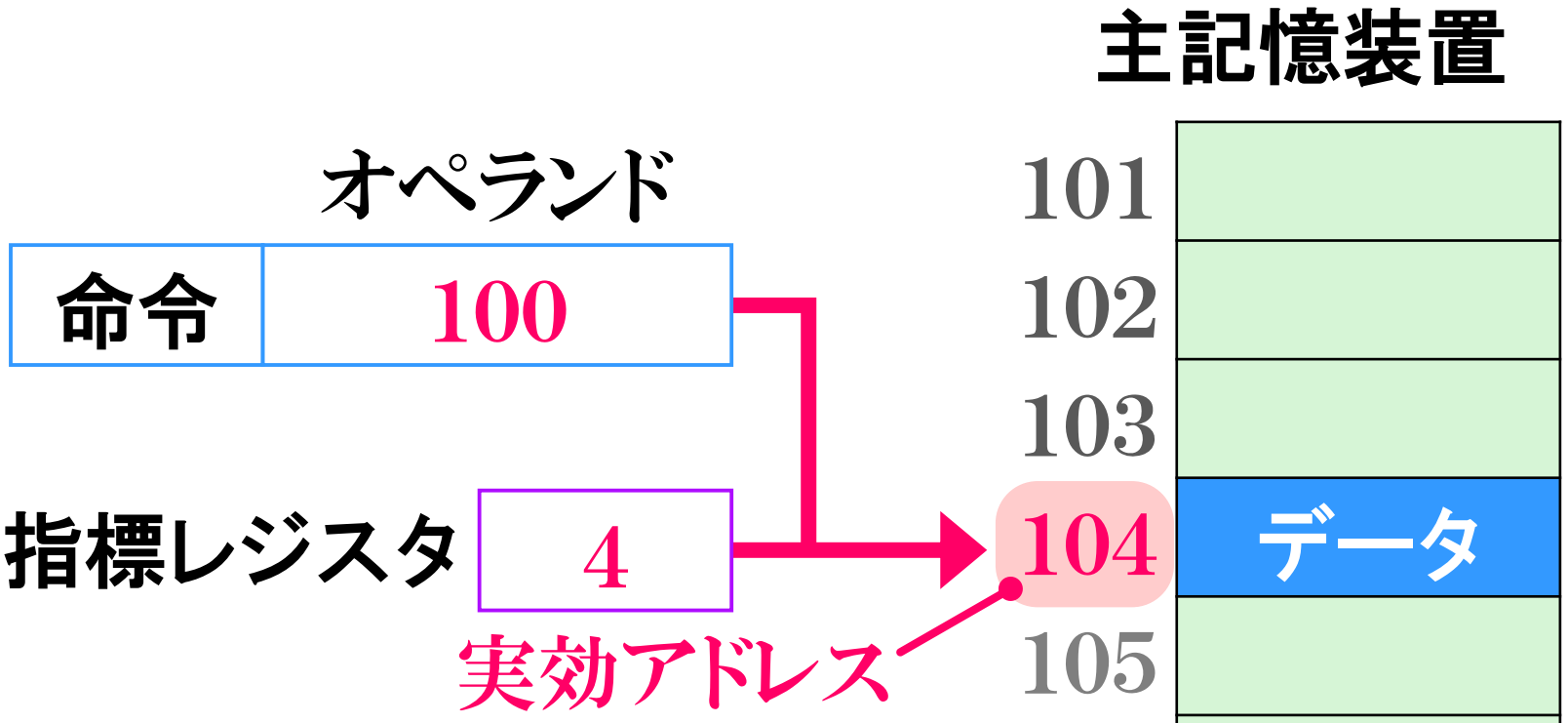
相対アドレス指定

オペランドのアドレス部の値に、PCの値を加算したものを実効アドレスとする方式



指標アドレス指定

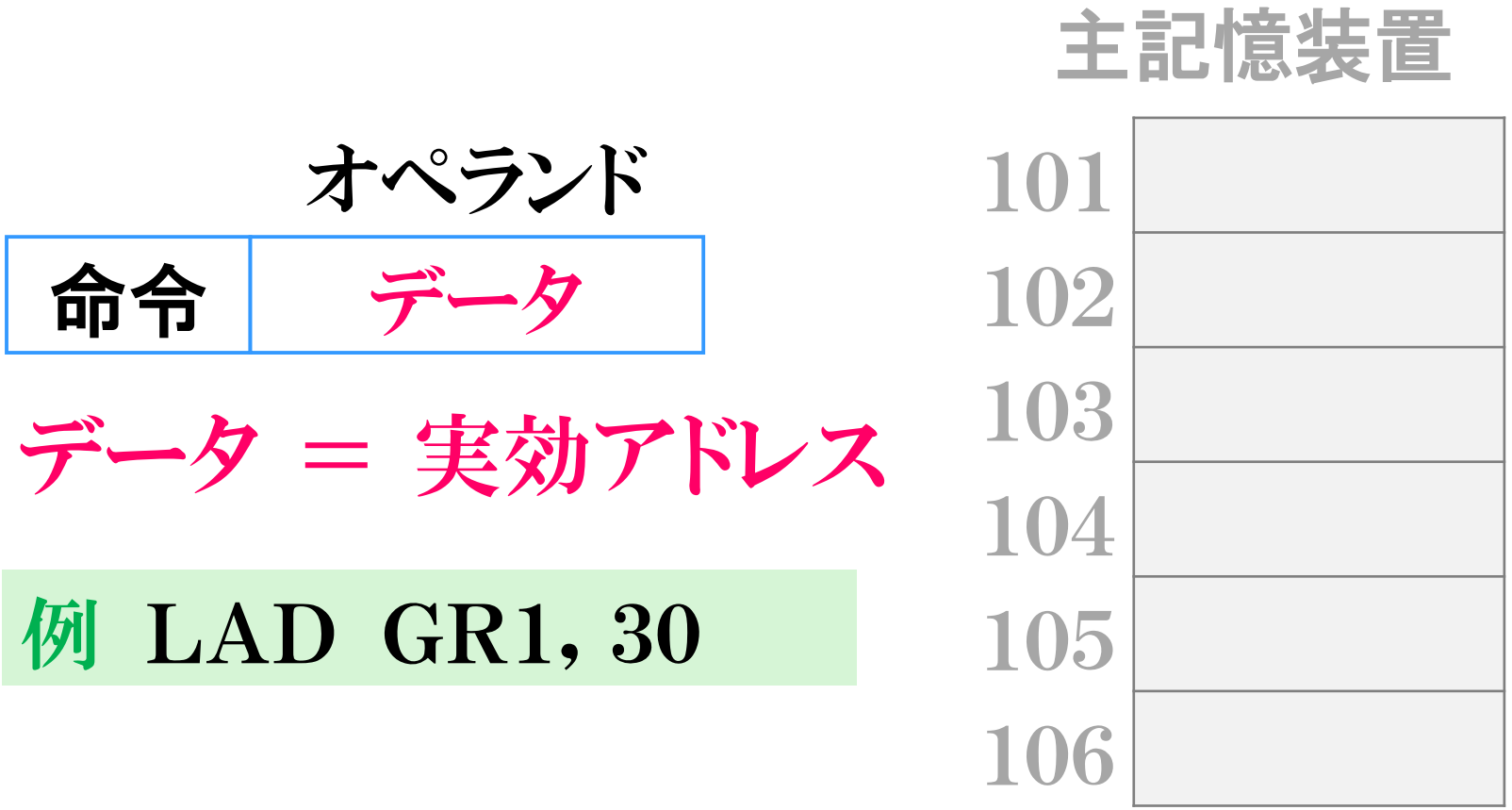
オペランドのアドレス部の値に、指標レジスタの値を加算したものを実効アドレスとする方式



例　LD　GR0, 100, GR1

即値アドレス指定

オペランドのアドレス部にデータそのものを格納する方式



データ ＝ 実効アドレス

例 LAD GR1, 30

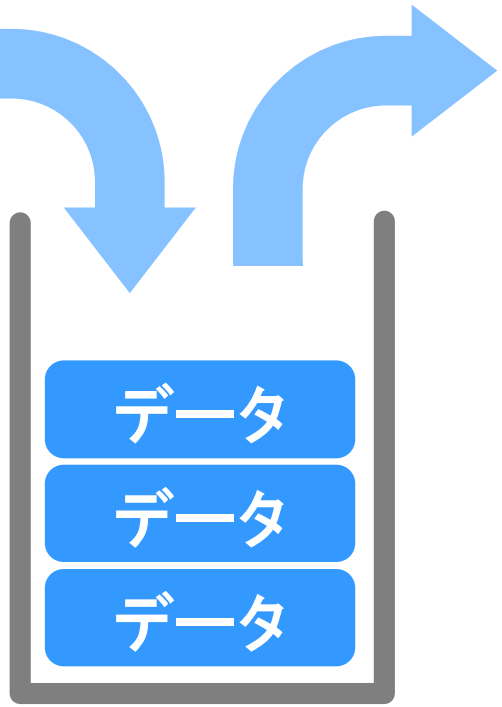
スタック

データを一時的に蓄えておくための主記憶装置上の記憶場所

後入れ先出し方式(LIFO)

Push
データを入れる

Pop
データを取り出す



PUSH／POP

PUSH　adr

adrをスタックへ入れる

PUSH　adr, x

(adr+レジスタx)をスタックへ入れる

POP　r1

スタックから取り出した値をレジスタr1へ書き込む

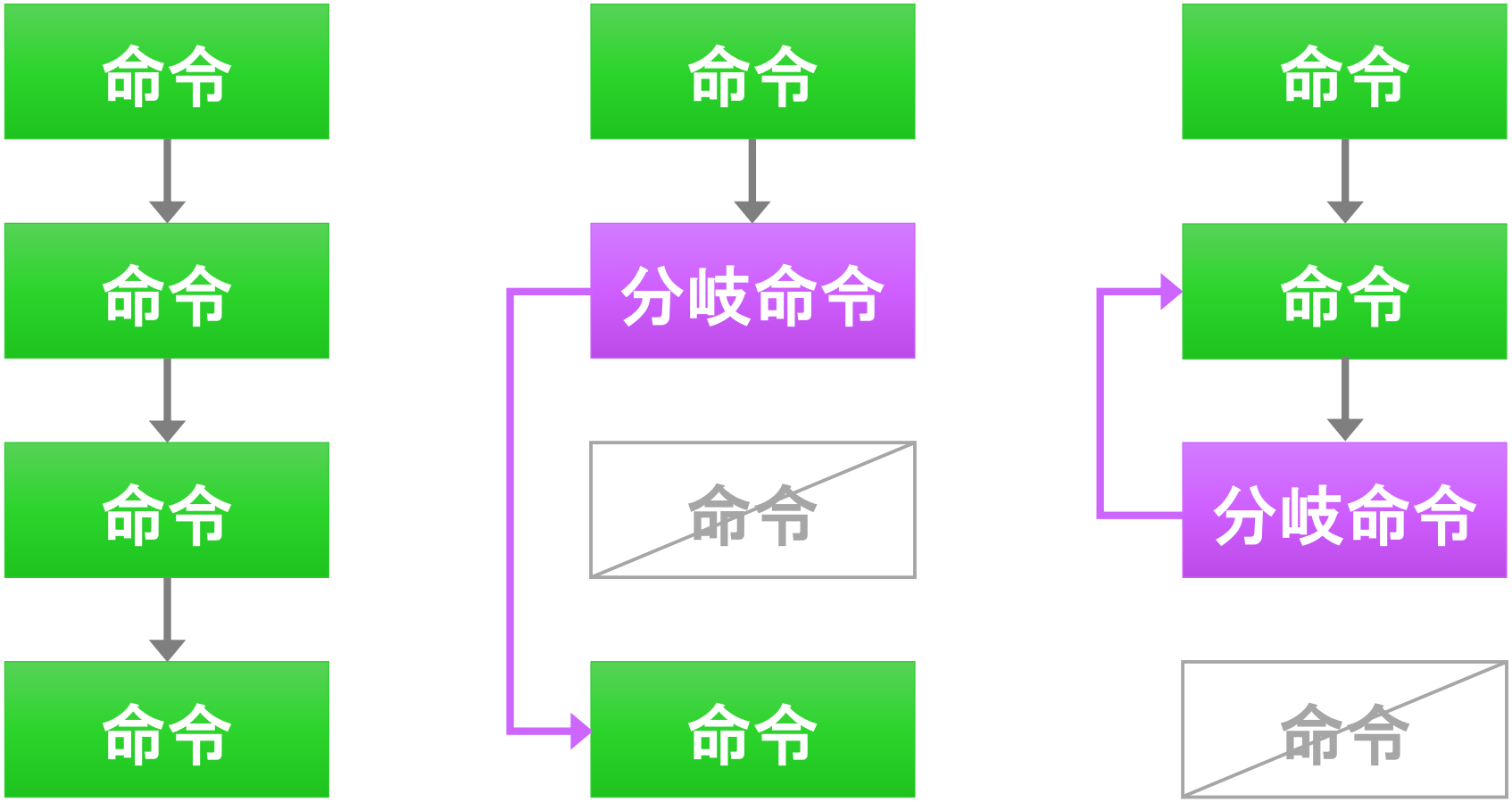
キュー

データを一時的に蓄えておく方式の一つ。
先入れ先出し方式（FIFO）。待ち行列。

Enqueue
データを入れる

Dequeue
データを取り出す

逐次実行と分岐



分岐命令の種類

- 無条件分岐
- 条件分岐
  - 負分岐
  - 正分岐
  - 零分岐
  - 非零分岐

無条件分岐

重要

プログラムの実行位置を移動する。

JUMP　adr

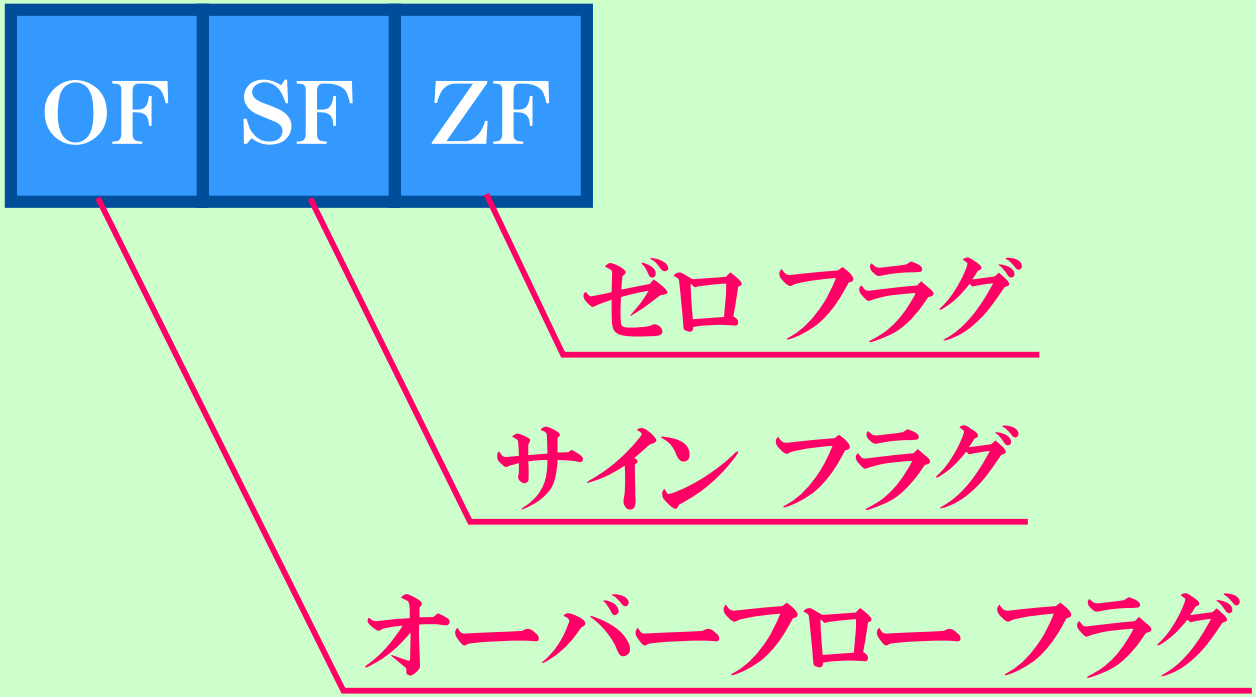
次の実行を、adr番地の命令へ移動する。

PC(プログラムカウンタ)の値をadrにする。

フラグレジスタ(FR)

重要

演算の結果やデータ転送の値の状態を示すレジスタ。3つのフラグ(各1bit)から成る。



フラグ

重要

- オーバーフローフラグ（OF）
  - 値が
    - 有効bitを越えたとき　→ 1
    - 有効bitを越えないとき → 0
- サインフラグ（SF）
  - 値が
    - 負のとき　→ 1
    - 0または正のとき → 0
- ゼロフラグ（ZF）
  - 値が
    - 0のとき　→ 1
    - 0以外のとき → 0

フラグの値

重要

	値		
	正の数	0	負の数
サインフラグ(SF)	0	0	1
ゼロフラグ(ZF)	0	1	0

算術比較命令

重要

2つの値の減算結果からFRを設定する。
減算結果の値は残さない。

CPA　r1，r2

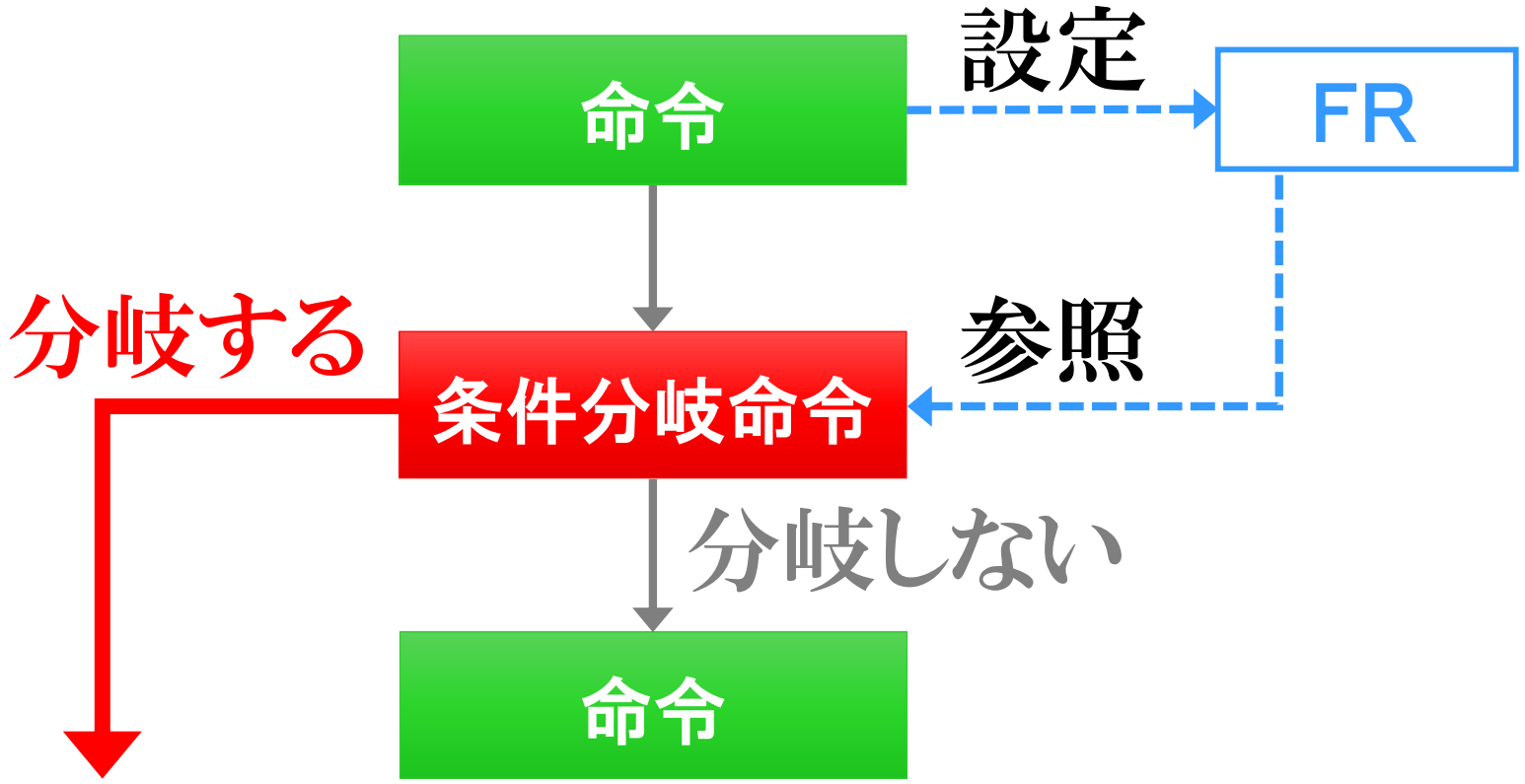
r1－r2の実行結果からFRを設定する。

CPA　r1，adr

r1－adr番地の値の実行結果からFRを設定する。

条件分岐命令

FRの値によって、分岐するか、しないかを決定する。



条件分岐命令

重要

JMI adr　負分岐

負（SF＝1）のときadr番地へ移動

JPL adr　正分岐

正（SF＝0，ZF＝0）のときadr番地へ移動

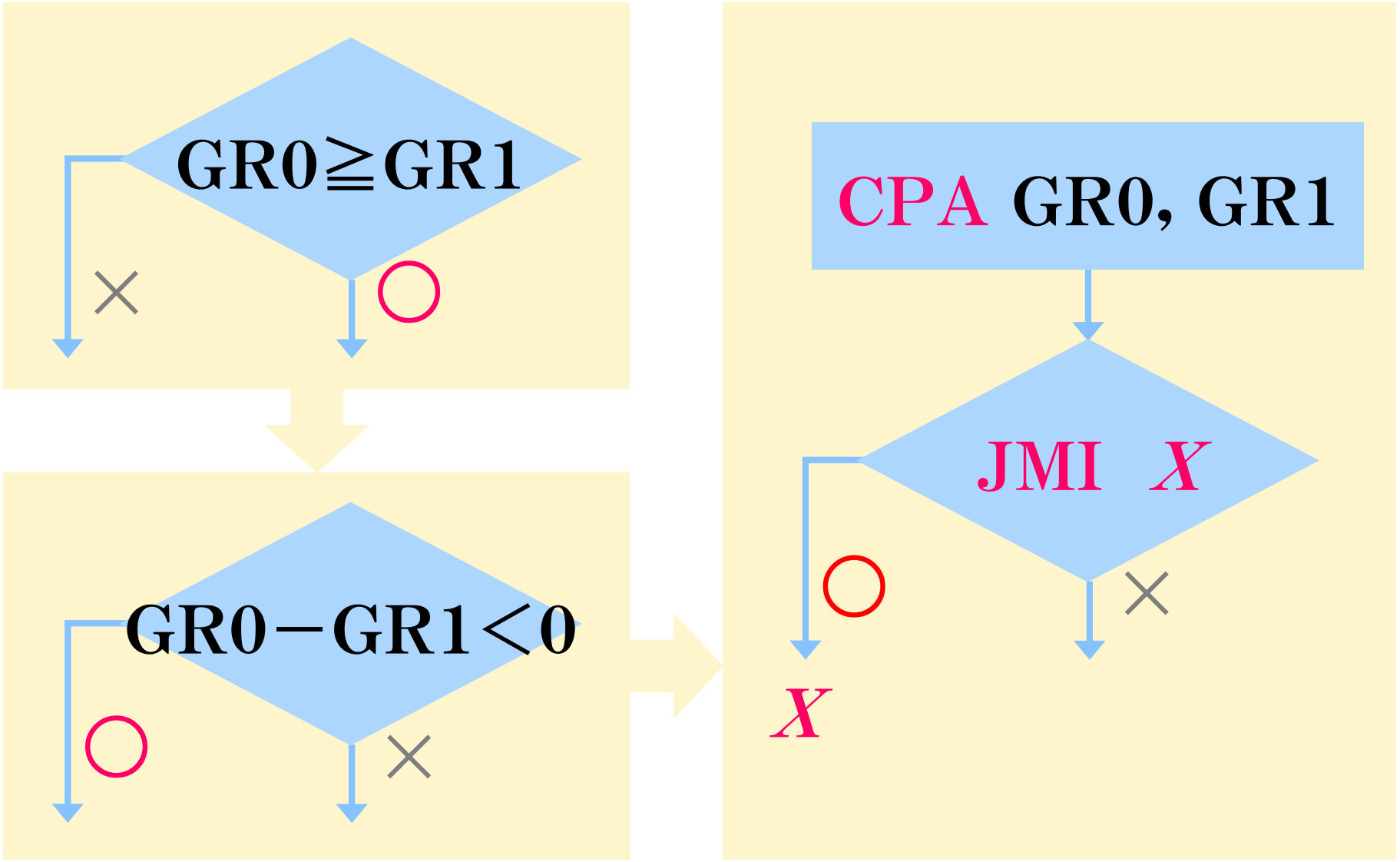
JZE adr　零分岐

0（ZF＝1）のときadr番地へ移動

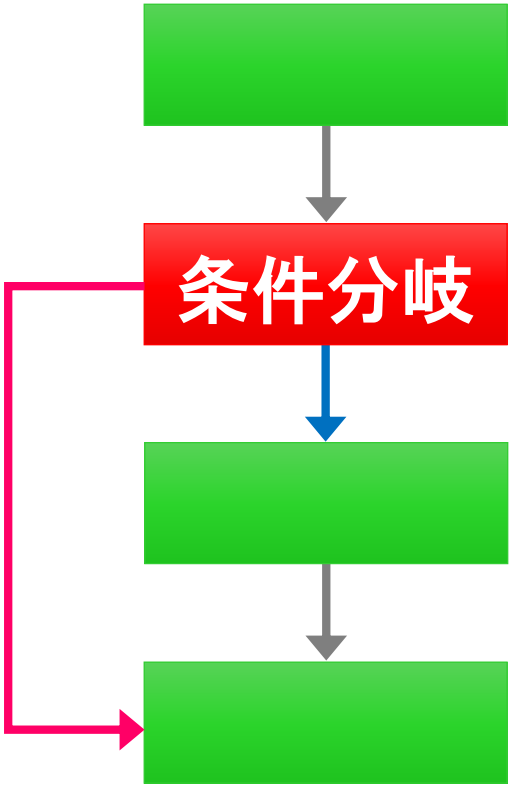
JNZ adr　非零分岐

0でない（ZF＝0）のときadr番地へ移動

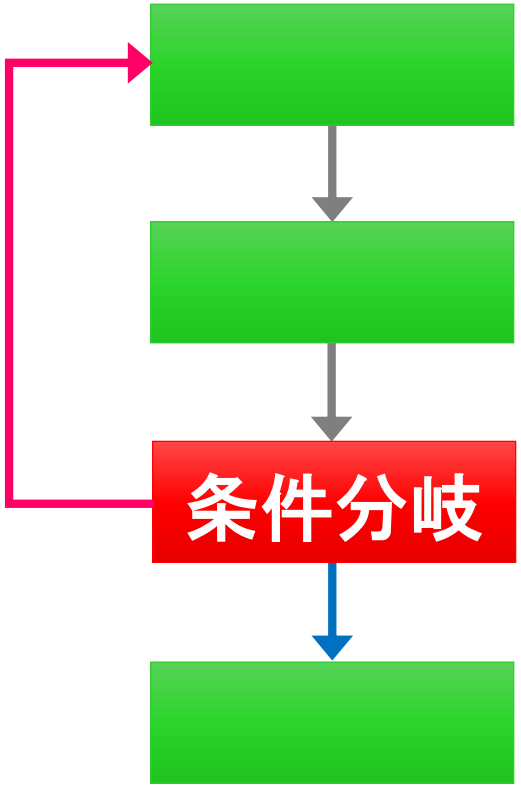
条件分岐の実現



条件分岐の使用

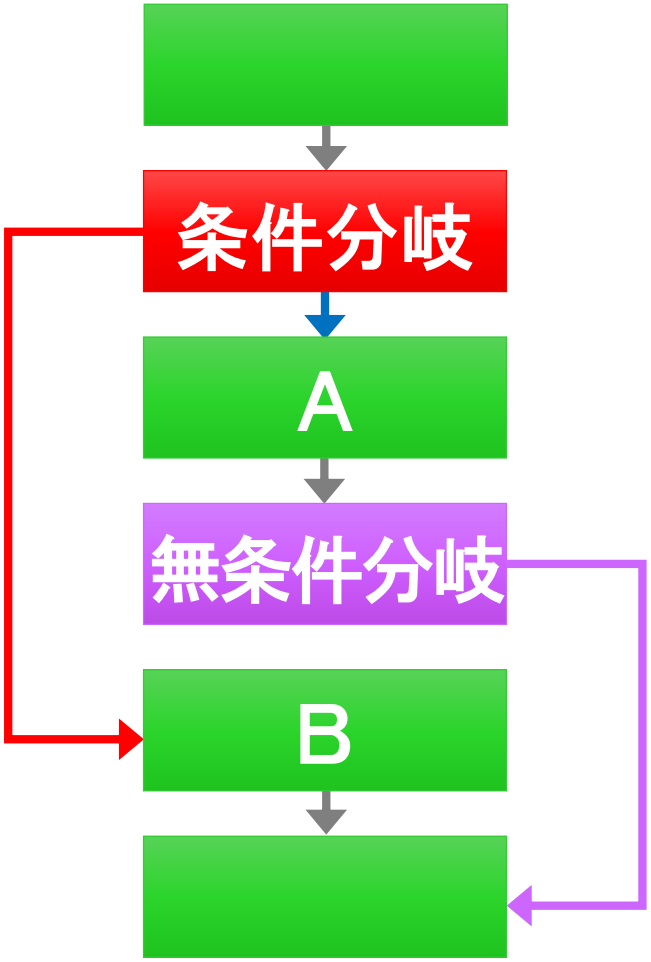


if文型

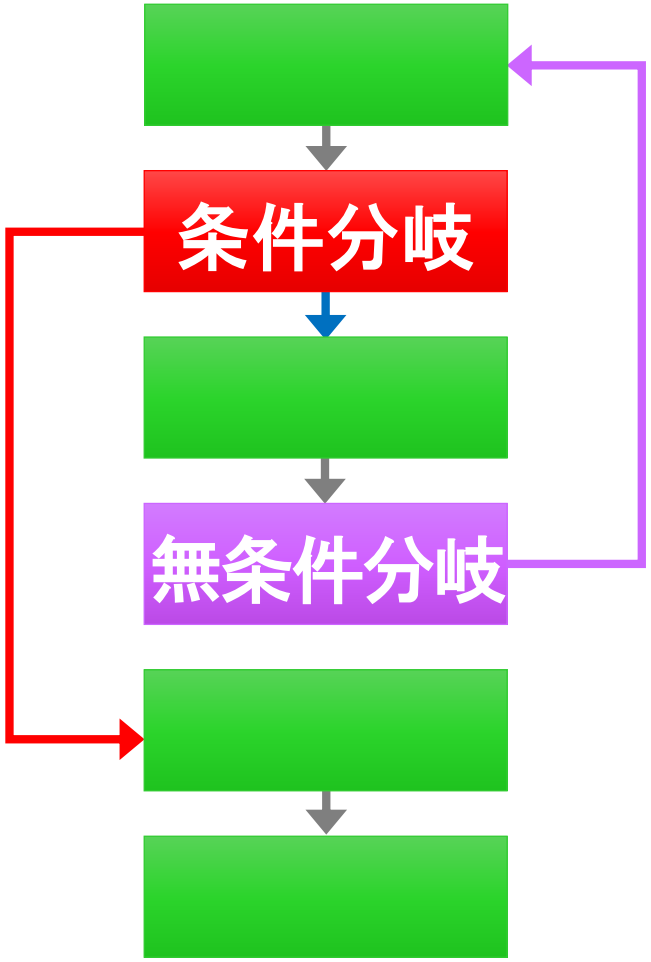


do-while文型

条件分岐と無条件分岐の使用



if-else文型



while文型

課題 3

✣ 配付プリント

提出日時：１月23日（水）講義開始前

定期試験　１月28日

CPUアーキテクチャ

✤ CISC　複雑命令セットコンピュータ

1つの命令で複雑な処理を実行できる。

組み込まれている命令数は多い。

✤ RISC　縮小命令セットコンピュータ

1つの命令は単純な処理だけを行う。

組み込まれている命令数は少ない。

※ CPUの進歩により、現在は明確な区別が無くなってきている。

CISCとRISCの特徴

	CISC	RISC
利点	小さいプログラムで、複雑な処理が行える。	命令の実行時間が短く、どれも同じ長さである。 CPUの回路構造が簡単になる。
欠点	命令の実行時間が長く、命令ごとに時間が違う。 CPUの回路構造が複雑になる。	複雑な処理をしたいとき、プログラムが大きくなる。
用途	パソコン用CPUなど	組み込み機器用小型マイコンなど

CPUの高速化法

- パイプライン処理
- マルチプロセッサ，マルチコア
- SIMD演算

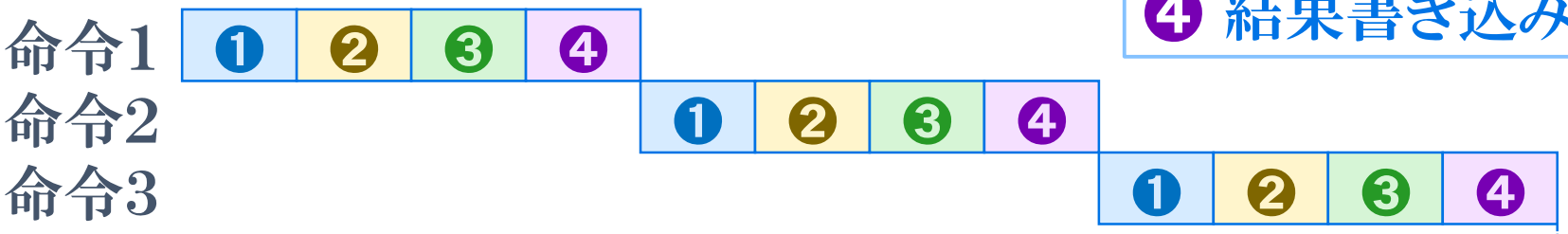
パイプライン処理

重要

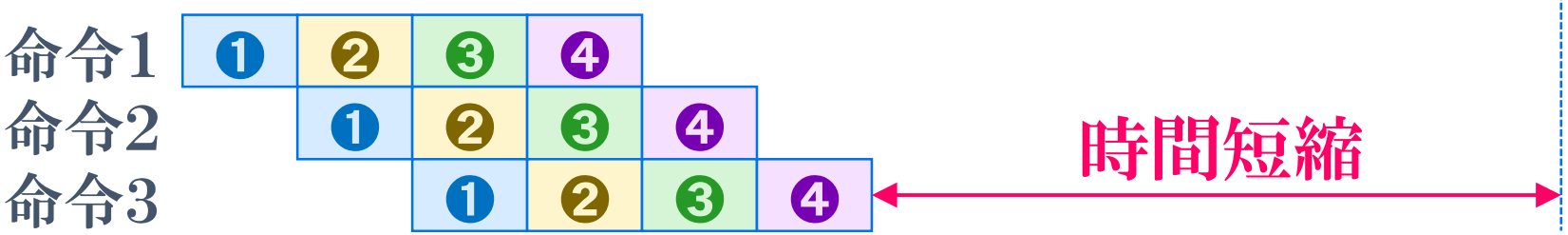
複数の命令を並行して実行することで、処理速度を上げる。

❶ 命令読み出し
❷ 命令解読
❸ 命令実行
❹ 結果書き込み

逐次処理



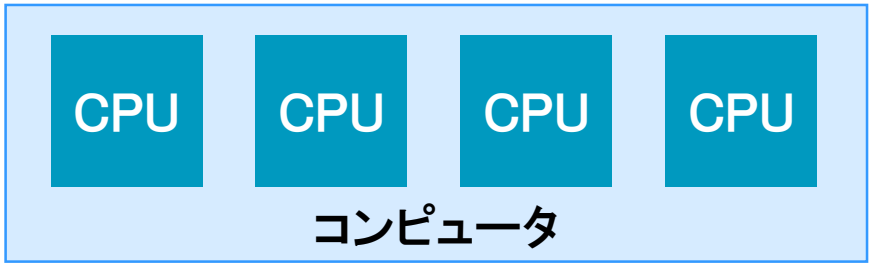
パイプライン処理



マルチプロセッサ／マルチコア

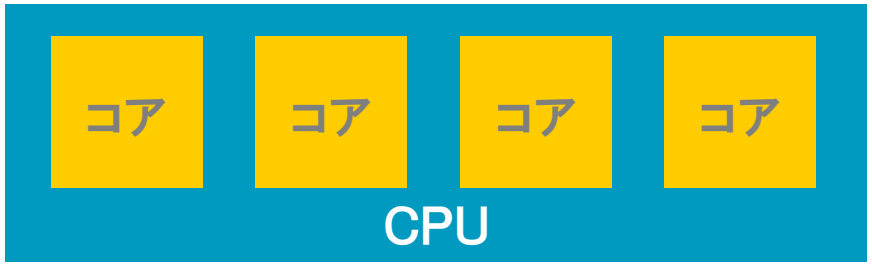
✤ マルチプロセッサ

1台のコンピュータの中に複数のCPUを搭載



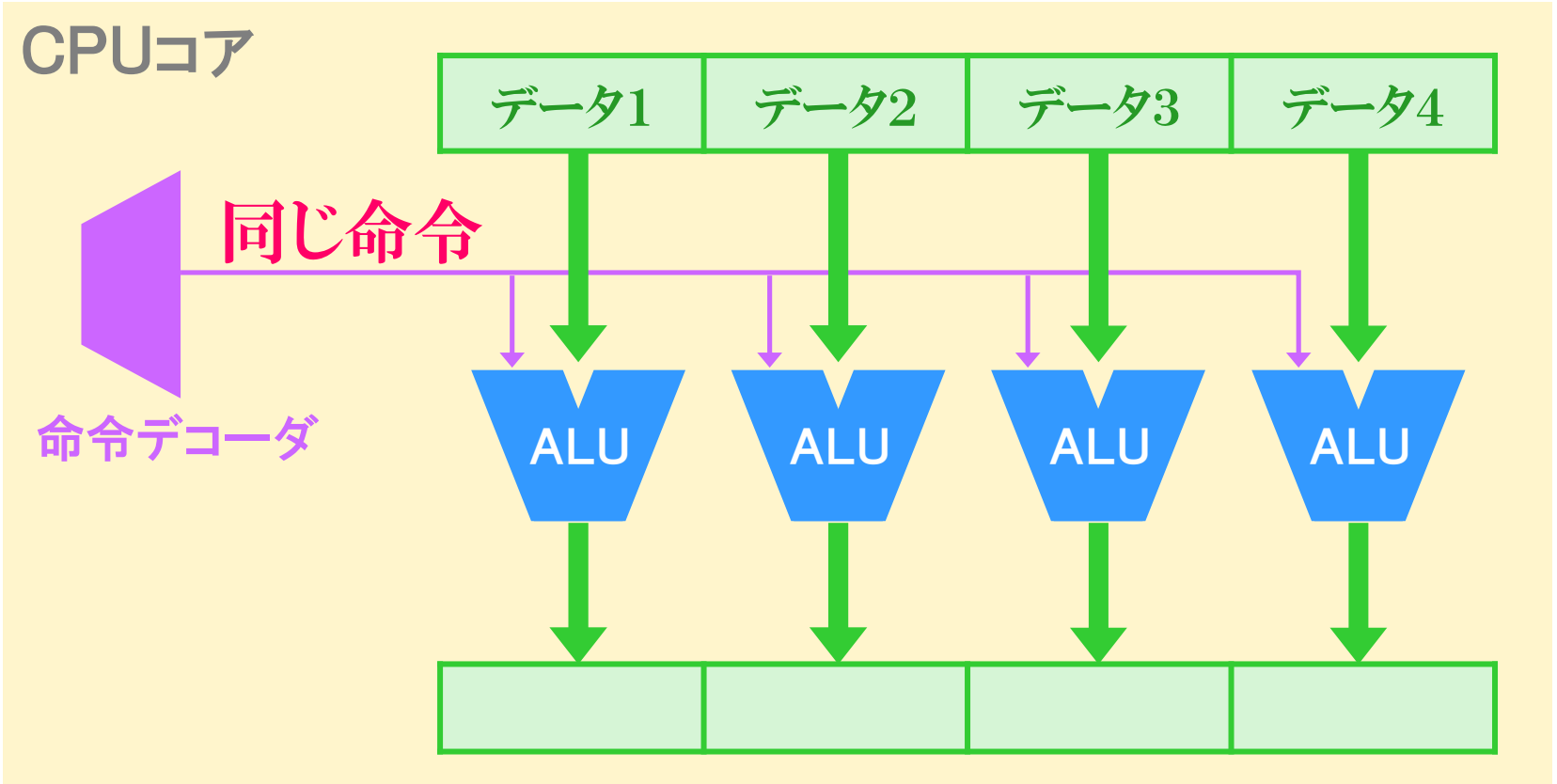
✤ マルチコア

1個のCPUの中に複数のCPUコアを搭載



SIMD方式

1つの命令で複数のデータを処理することにより、処理速度を上げる。



CPUの性能評価

重要

- 平均命令実行時間

  1命令の実行に要する時間（秒）の平均値

- MIPS

  1秒間に実行できる命令数（百万単位）

- FLOPS

  1秒間に実行できる浮動小数点数演算の回数

定期試験(※予定)

日時　2019年1月28日　9時10分

場所　C0241講義室

出題範囲

- CPUの構成要素と働き。命令サイクル
- アセンブリ言語の記述。実効アドレス
- フラグと分岐処理
- CPUの高速化、CPUの性能評価
- 中間試験範囲から基礎的問題