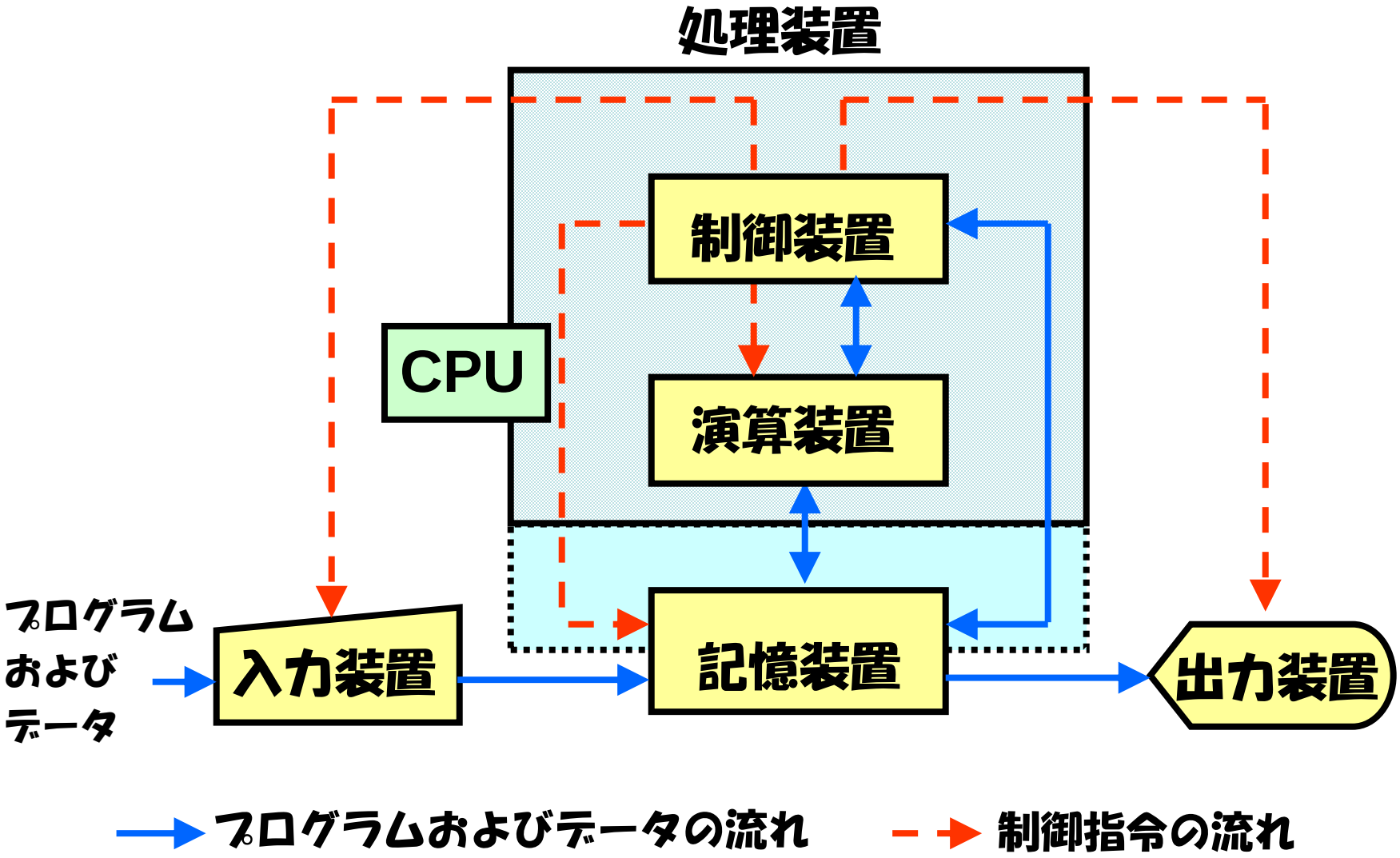
コンピュータアーキテクチャ
2024

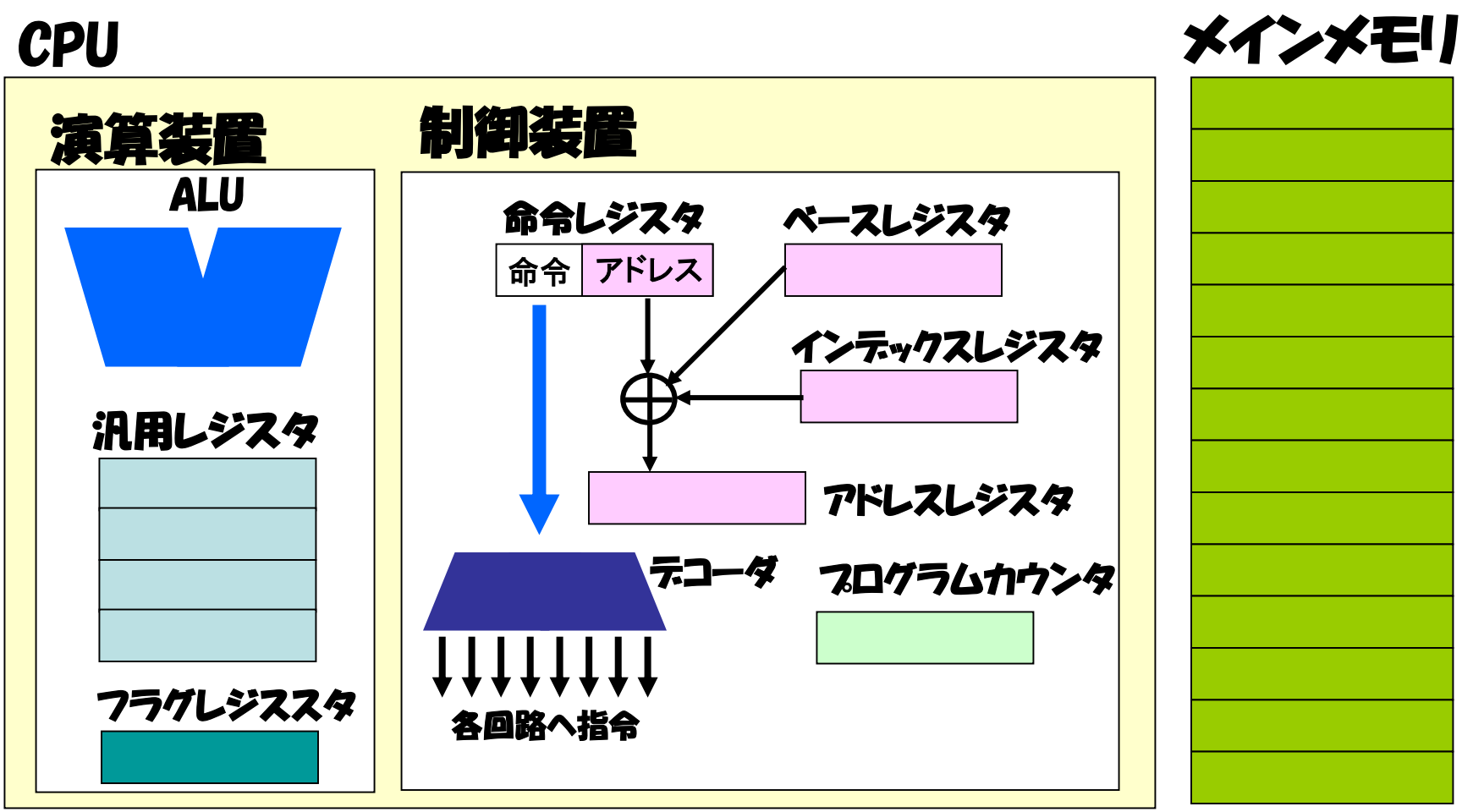
CPUの内部動作

堤　利幸

コンピュータの構造と5大装置



CPUの内部構造


CPU
メインメモリ
演算装置
制御装置
ALU
汎用レジスタ
フラグレジススタ
命令レジスタ
命令
アドレス
ベースレジスタ
インデックスレジスタ
アドレスレジスタ
デコーダ
プログラムカウンタ
各回路へ指令

制御装置

- デコーダ
  - 命令の解釈を行う回路
- プログラムカウンタ
  - 実行中あるいは次に実行する命令が格納されている番地を保持する
- 命令レジスタ
  - 取り出した命令を保持する
- ベースレジスタ
  - 主記憶装置に置かれるプログラムの先頭番地を保持する
- インデックスレジスタ
  - インデックスアドレス指定で用いる指標値を保持する
- アドレスレジスタ
  - 命令のアドレス部，ベースレジスタ，インデックスレジスタの値を基にして求めた主記憶の番地を保持する

演算装置

- 汎用レジスタ
  - プログラムで使い方を自由に決められるレジスタで，0，1，2．．のように番号で指定する
  - 被演算数や演算結果を一時的に保持する

- フラグレジスタ(ステータスレジスタ)
  - 演算結果の正負や桁上がり等のCPUの状態情報を保持する

- 算術論理演算回路(ALU: Arithmetic-Logic Unit)
  - 命令を解読したデコーダの指示に従い，レジスタを利用しながら実際に四則演算を行う回路(AU: Arithmetic Unit)と論理演算を行う回路(LU: Logic Unit)

    算術論理演算回路　ALU

    ＝ 算術演算回路　AU　(+, -)

    ＋ 論理演算回路　LU　(AND, OR, ...)

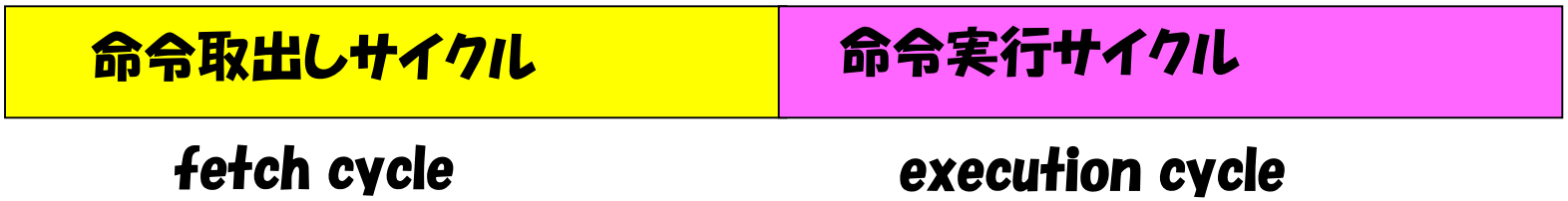
命令形式

Instruction	Operation code Opecode	Operand
命令語	命令部	アドレス部
「・・を ～せよ」	「～せよ」	「・・番地のデータを」

命令サイクルと6つのステージ

CPUの動作は，プログラム（機械語命令の集まり）をメインメモリに格納し，
その先頭アドレスをプログラムカウンタにセットすることから始まります．

命令の実行制御は，大別して
メインメモリから機械語命令を取り出す「CPU内部への命令取出しサイクル」と，
機械語命令を解読（デコード:decode）後実行し結果をメインメモリに
書き込む「CPU内部での命令実行サイクル」からできている．
（サイクルとは「処理段階」の意．）



この２つのサイクルを繰り返しながら処理を続けることにより，
プログラムを実行します．
２つのサイクルを合わせて「命令サイクル」と言い，命令サイクル
はさらに６つの処理から成り立っており，この処理のことを
「ステージ」と言います．

命令実行するためのCPUの内部動作

1アドレス命令による足し算プログラムの実行例

メモリ中にプログラム（命令とデータ）が格納されている．

[命令]

0000番地	LD	100	; (Acc) ← (100)
0001番地	ADD	200	; (Acc) ← (Acc) + (200)
0002番地	ST	300	; (Acc) → (300)

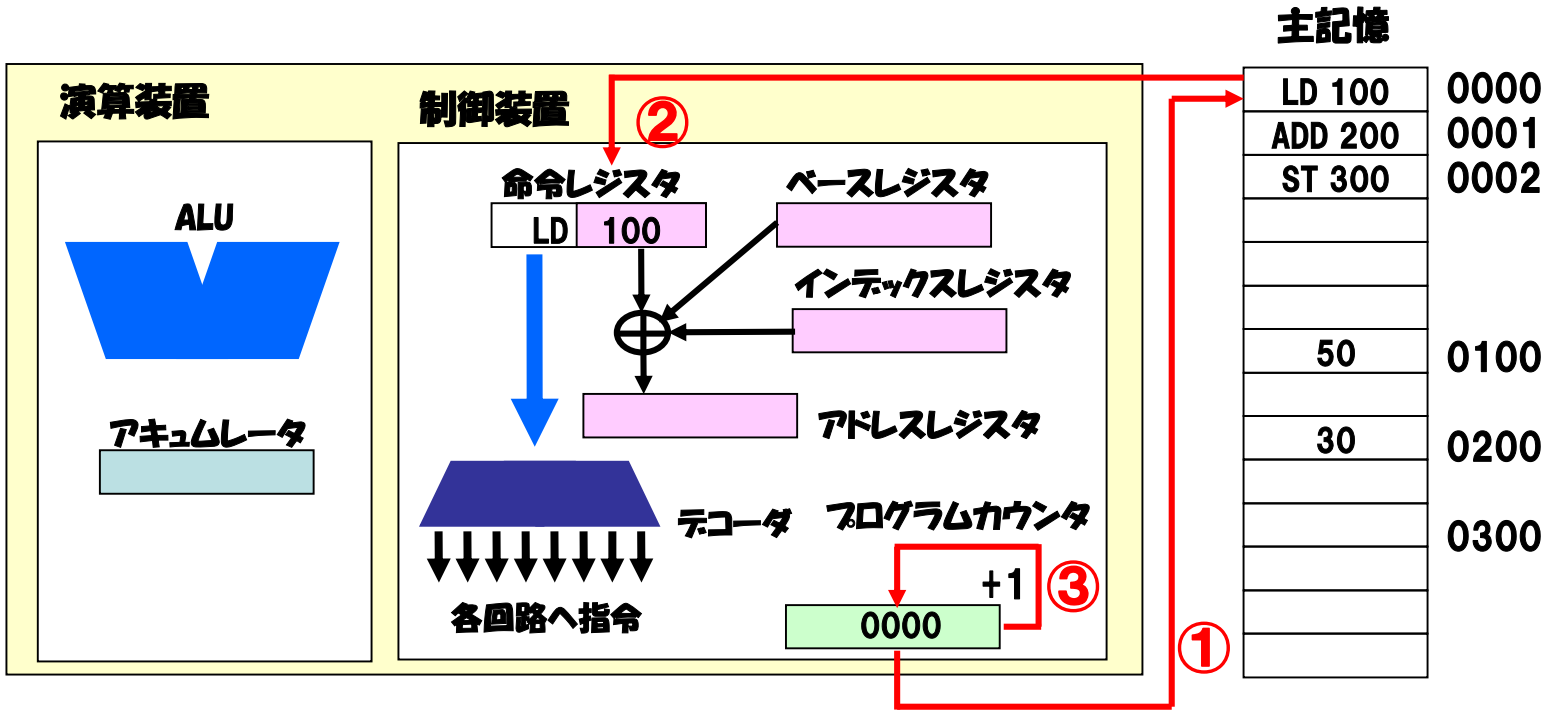
[データ]

0100番地	50
0200番地	30

命令実行するためのCPUの内部動作(1)

命令取出しサイクル（Instruction Fetch Cycle）

① プログラムカウンタで取出す命令の番地を指定
② 命令を取出す
③ プロラムカウンタに命令の長さ（＝語長）を加える
- 次の命令の番地を指定することになる.



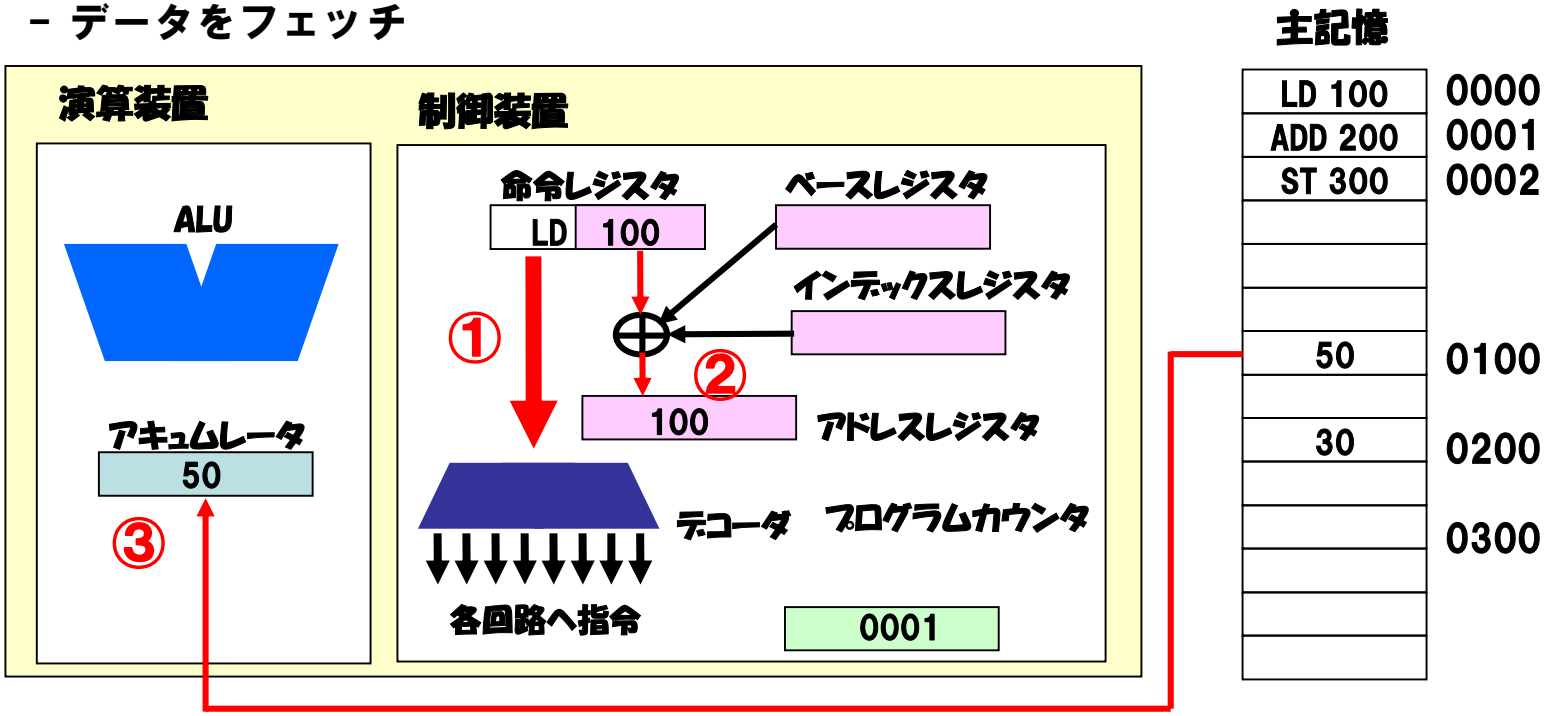
命令実行するためのCPUの内部動作(2)

命令実行サイクル（Instruction Execution Cycle)

① 命令をデコーダで解読する
- LD(ロード命令）であることが分かる

② 読み出すべきデータのアドレス番地を求める
- アドレス部の値がそのまま主記憶のアドレスとなる（直接番地指定)

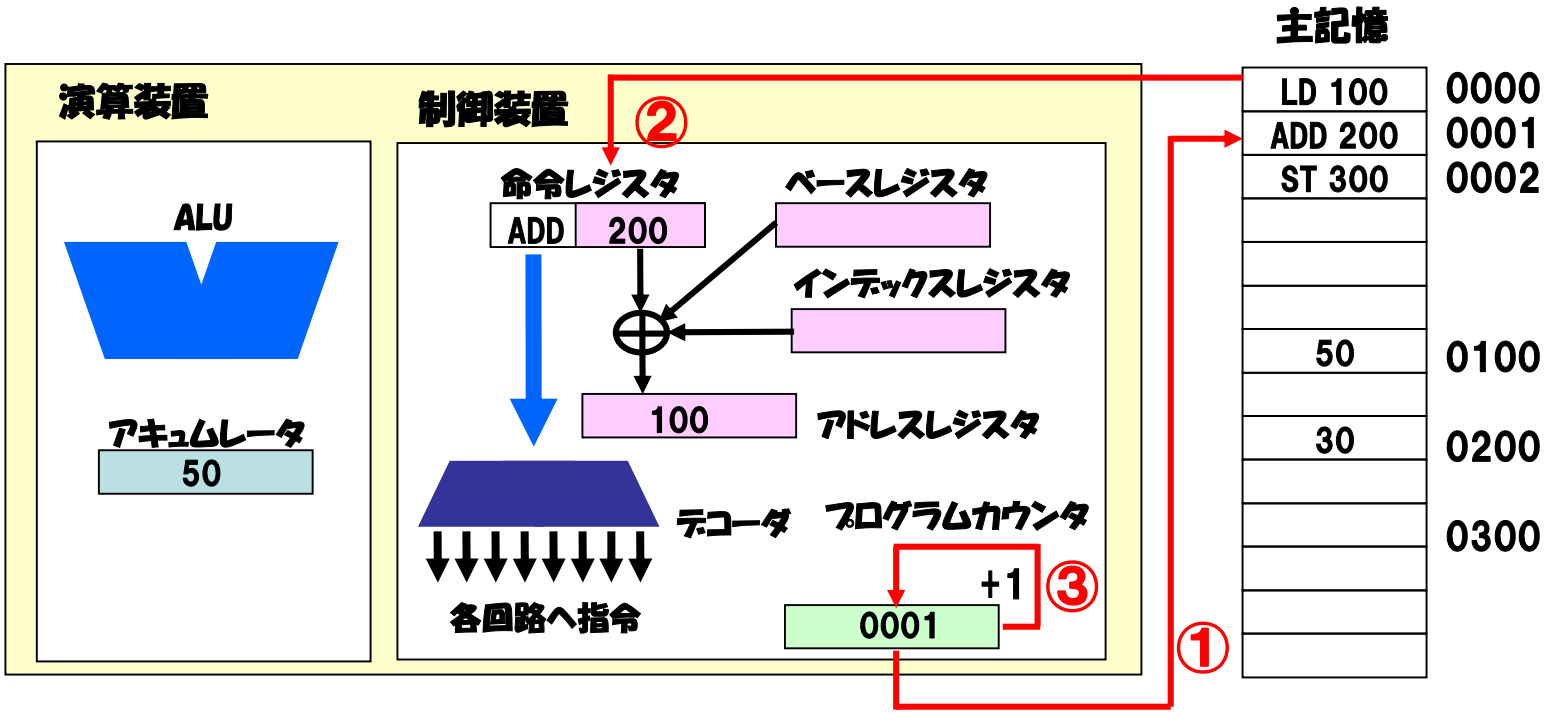
③ デコーダ（シーケンサ）が，指定されたアドレスのデータをアキュムレータに取出す
- データをフェッチ



命令実行するためのCPUの内部動作 (3)

命令取出しサイクル（Instruction Fetch Cycle）

① プログラムカウンタで取出す命令の番地を指定
② 命令を取出す
③ プロラムカウンタに命令の長さ（＝語長）を加える
- 次の命令の番地を指定することになる.



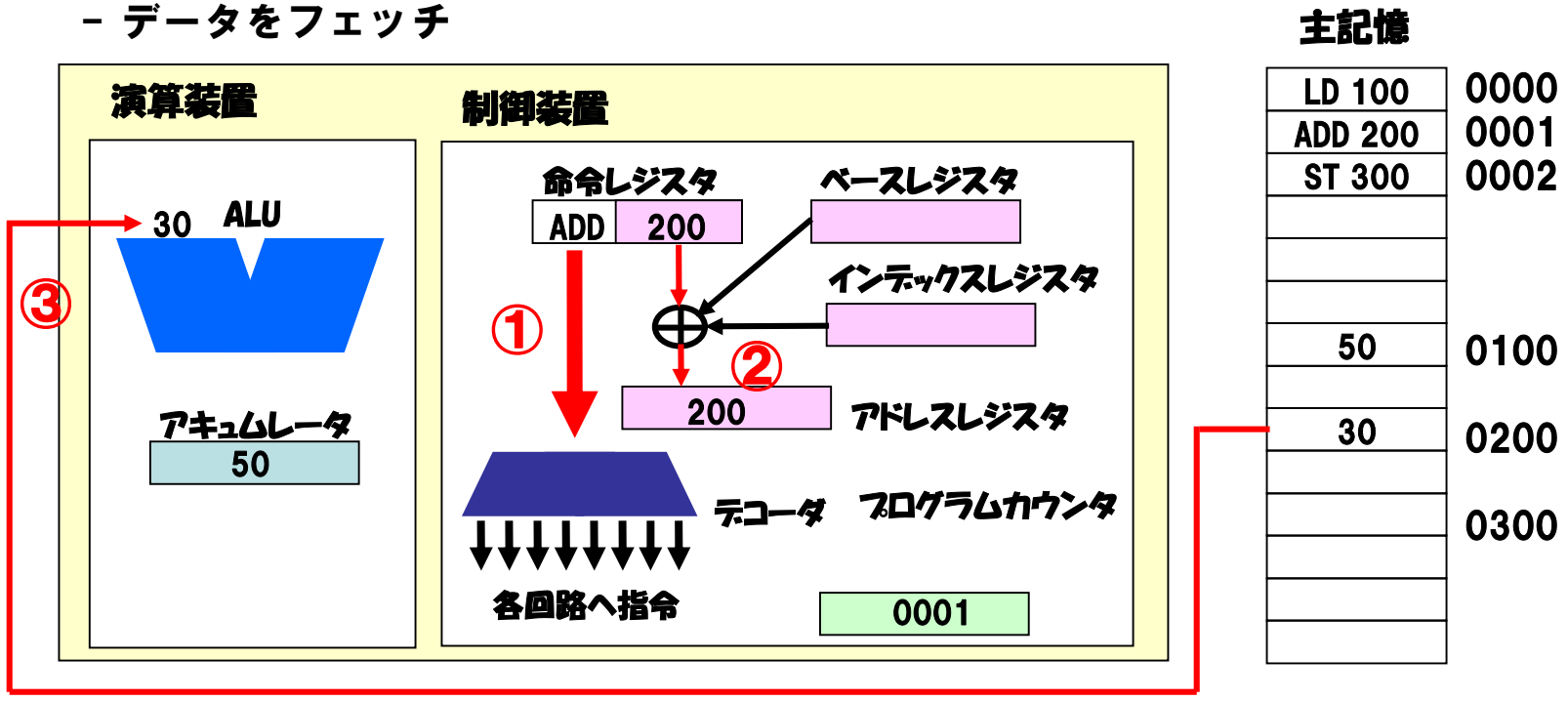
命令実行するためのCPUの内部動作(4)

命令実行サイクル（Instruction Execution Cycle)

① 命令をデコーダで解読する
- ADD(加算命令）であることが分かる

② 加算すべきデータのアドレス番地を求める
- アドレス部の値がそのまま主記憶のアドレスとなる（直接番地指定)

③ デコーダ（シーケンサ）が，指定されたアドレスのデータを加算器に取出す
- データをフェッチ



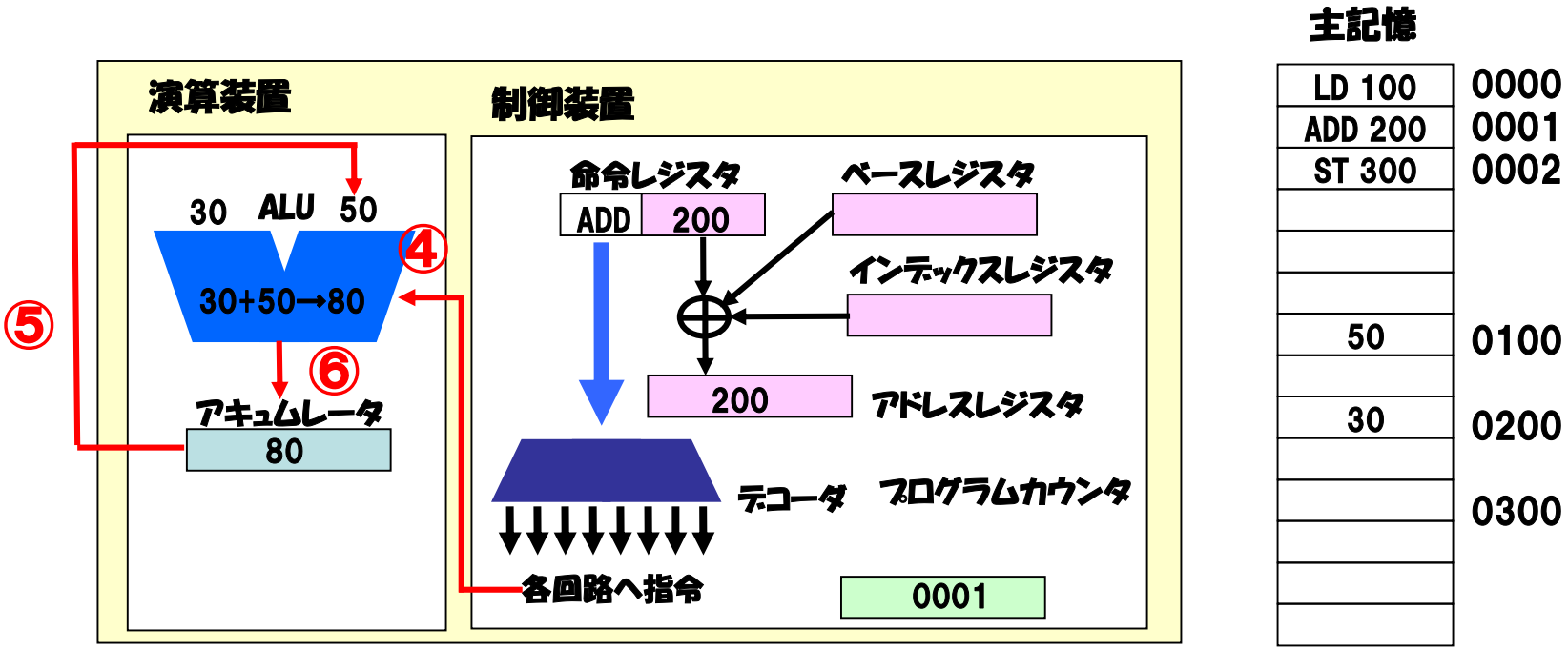
命令実行するためのCPUの内部動作(5)

命令実行サイクル（Instruction Execution Cycle）

④　デコーダ（シーケンサ）がALUの中のデータとアキュムレータのデータを加算するように命令

⑤　アキュムレータのデータをALUに読み出し演算を実行

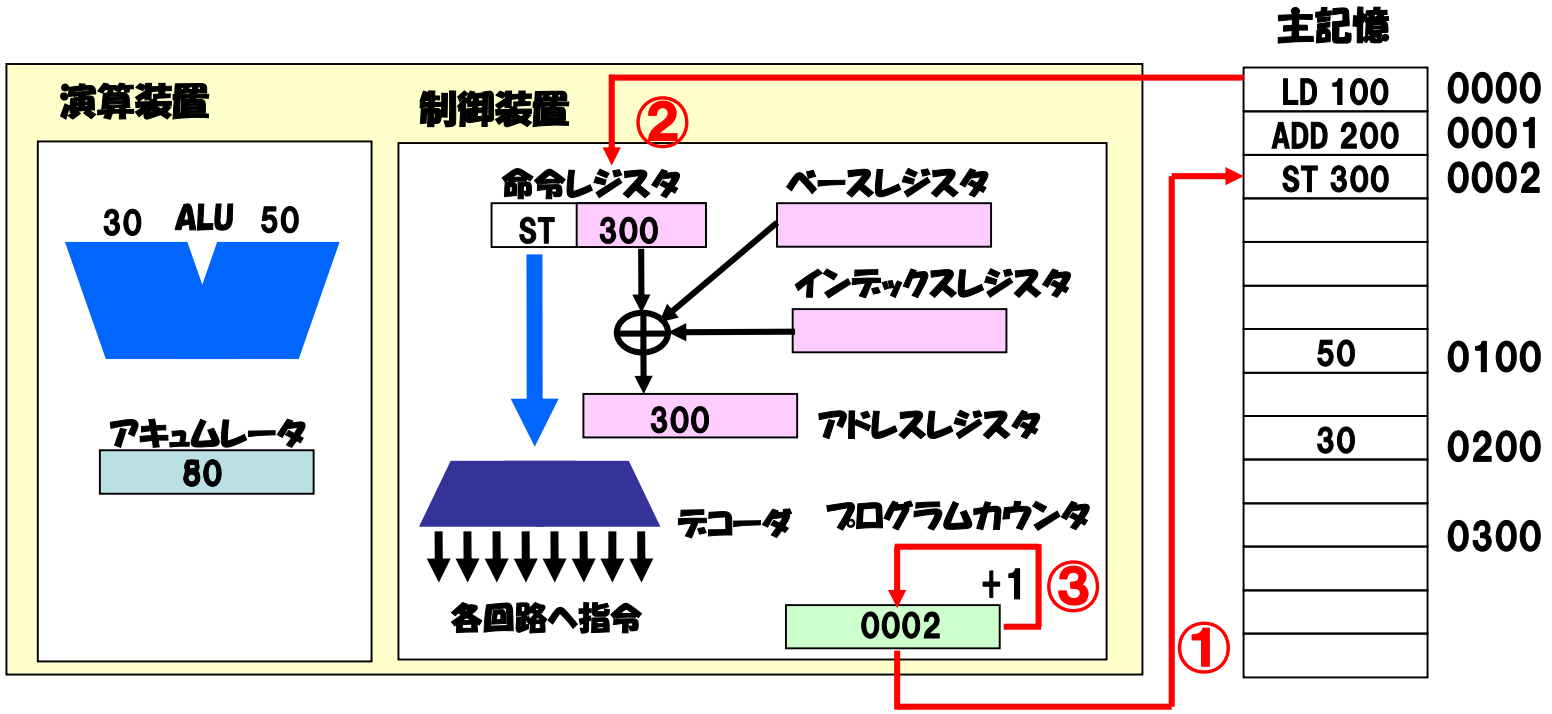
⑥　演算結果をアキュムレータに格納



命令実行するためのCPUの内部動作(6)

命令取出しサイクル（Instruction Fetch Cycle）

① プログラムカウンタで取出す命令の番地を指定
② 命令を取出す
③ プロラムカウンタに命令の長さ（＝語長）を加える
- 次の命令の番地を指定することになる.



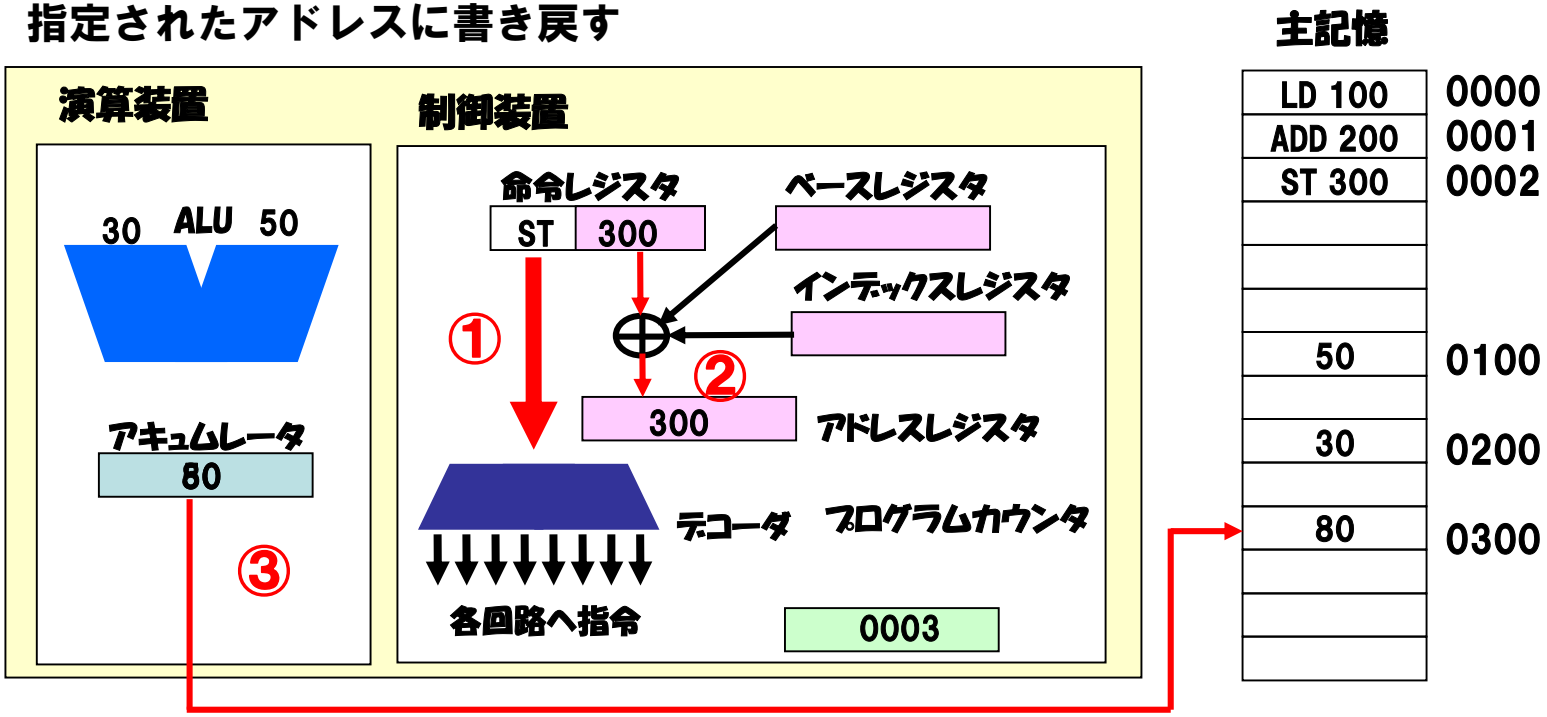
命令実行するためのCPUの内部動作(7)

命令実行サイクル（Instruction Execution Cycle)

① 命令をデコーダで解読する
- ST(ストア命令）であることが分かる

② ストアすべきデータのアドレス番地を求める
- アドレス部の値がそのまま主記憶のアドレスとなる（直接番地指定)

③ デコーダ（シーケンサ）が，アキュムレータの内容を指定されたアドレスに書き戻す



コンピュータの内部動作

(1)コンピュータの構造を5大装置（機能）を図示して説明しなさい.

(2) CPUの内部構造を図示し，CPU内の各ユニットが何をするのかを説明しなさい.