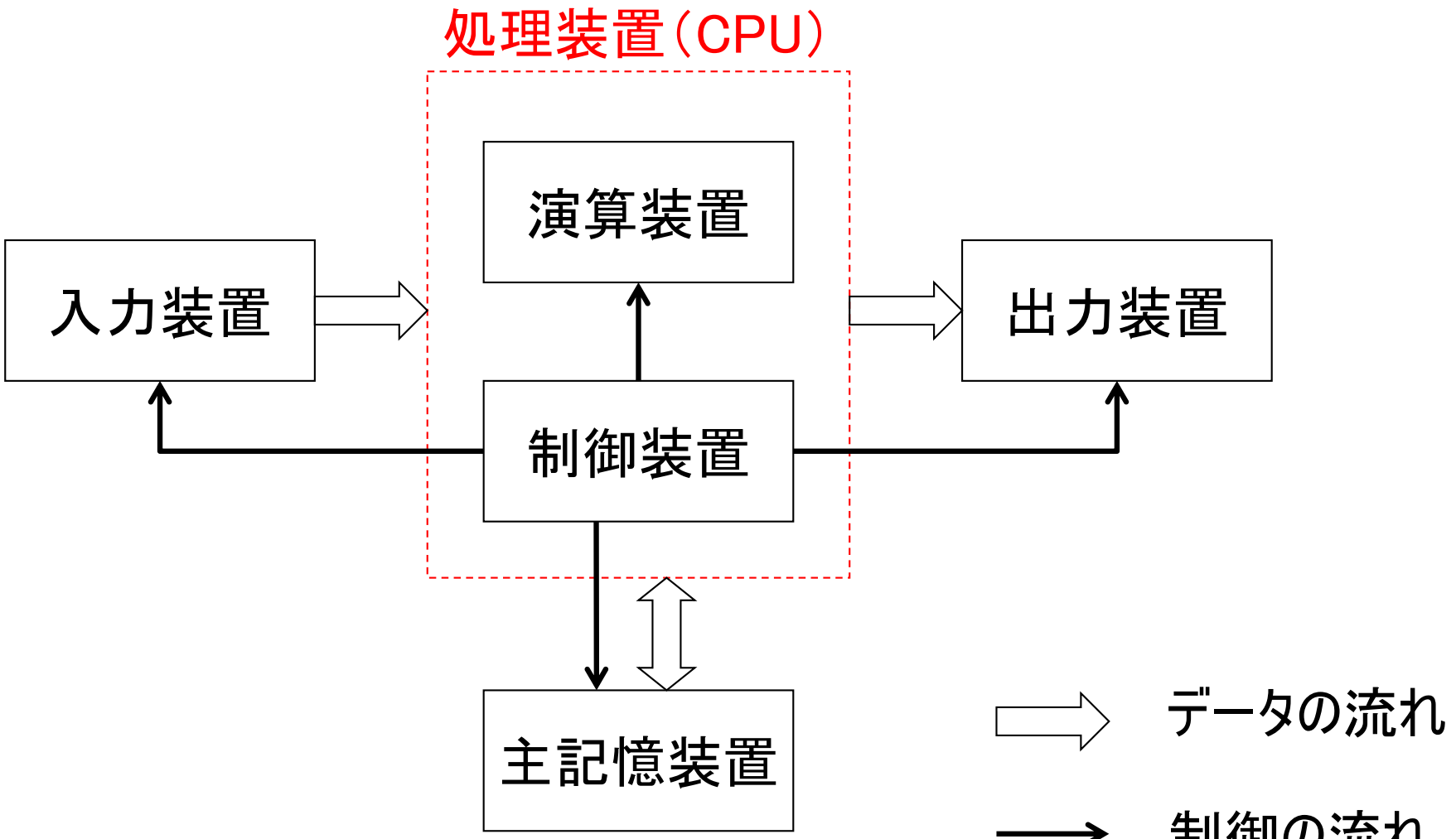
第8章「コンピュータシステム」

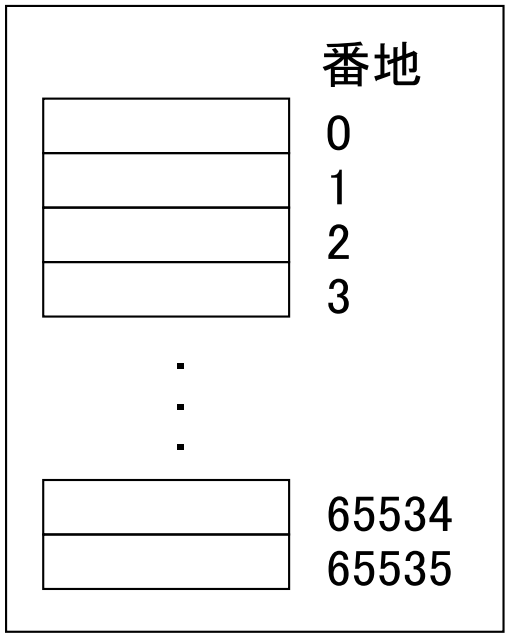
- コンピュータの構成
- 機械語プログラム
- CPUの構造
- レジスタの役割
- アドレス修飾
- 命令サイクル，取り出し・実行サイクル

コンピュータシステムの構成



主記憶(or 主メモリ)(装置)

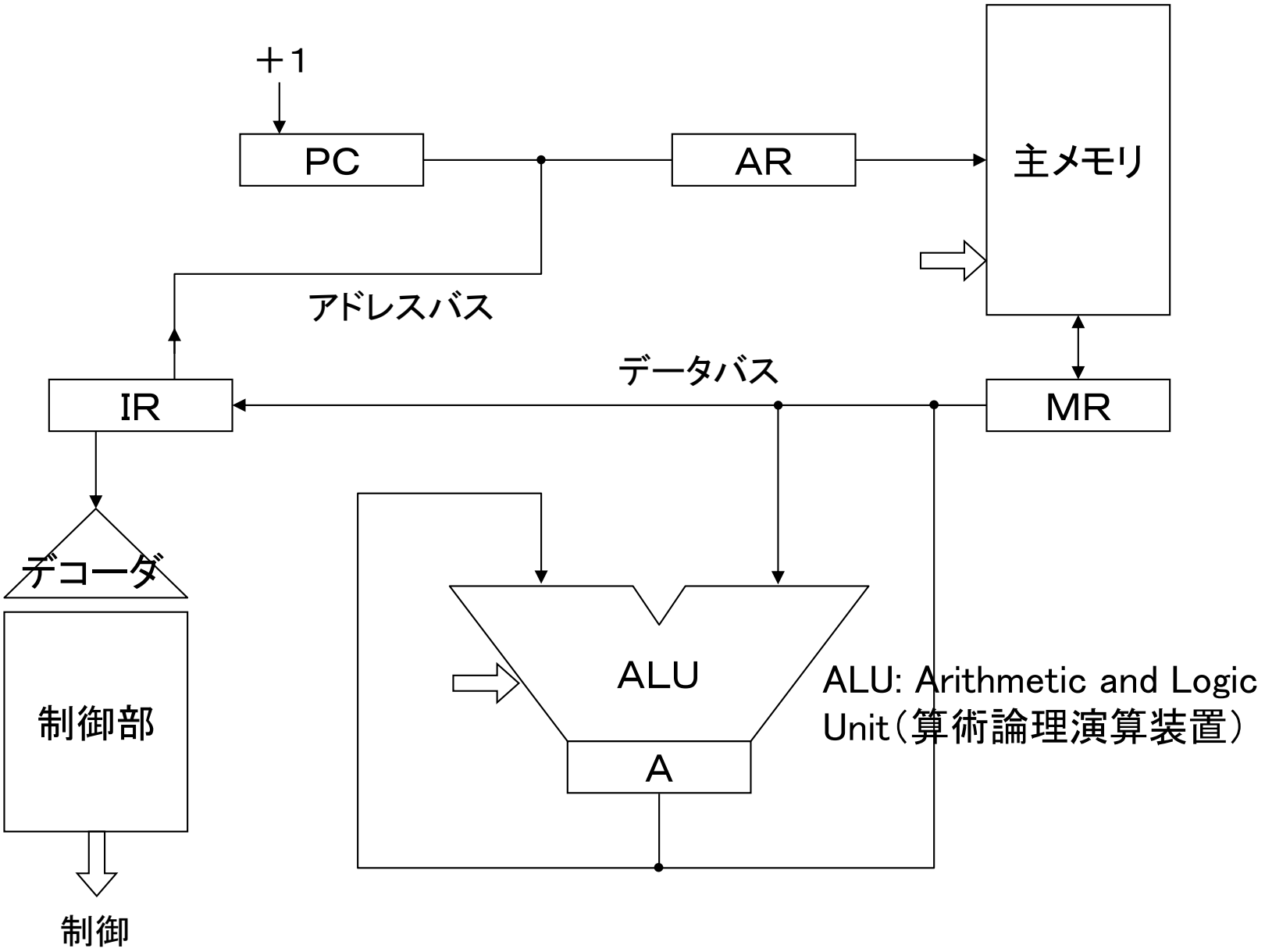
- データとプログラムを格納する記憶域
  - プログラム内蔵(stored program)方式(=ノイマン型)
- 1語単位でアドレス(番地)が振られており，アドレスを指定して1語単位で情報の書き込み・読み出しを行う
- 全記憶量＝アドレス広さ×語長
  - 例) アドレス 0～2^{12}，1語16ビット
) 記憶量 = 2^{12} × 16 ビット



アキュムレータ型CPUと汎用レジスタ型CPU

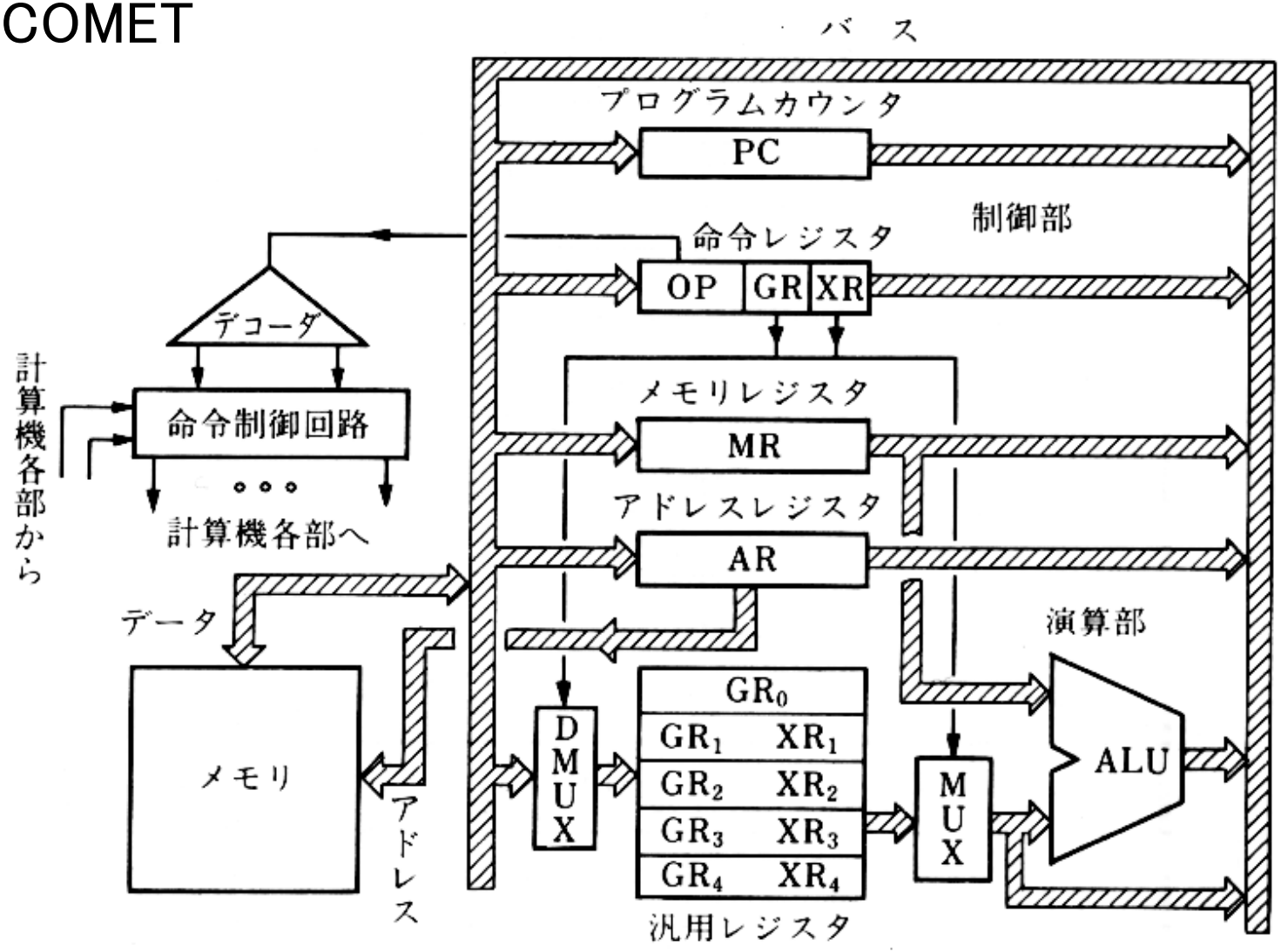
- アキュムレータ（accumulator）型CPU
  - 演算の結果が，常にひとつのレジスタ（アキュムレータ）を経由する
- 汎用レジスタ型CPU
  - 演算時，同じように扱えるレジスタが複数ある
  - 演算は複数あるレジスタとメモリの値の間で行い，結果をどれかのレジスタに格納する

アキュムレータ型CPUの構造



汎用レジスタ型CPUの構造

参考）COMET



レジスタ

本講義での表記	名称
IR	命令レジスタ
AR	メモリアドレスレジスタ
MR	メモリ(データ)レジスタ
PC	プログラムカウンタ
A	アキュームレータ
	フラグレジスタ
SR	状態(ステータス)レジスタ
	汎用レジスタ
	指標(インデックス)レジスタ

プログラマは直接操作はしない

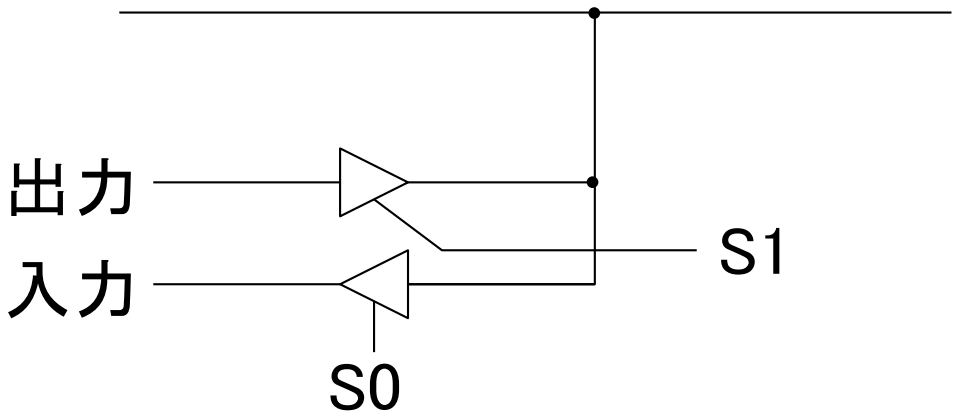
プログラマが操作する

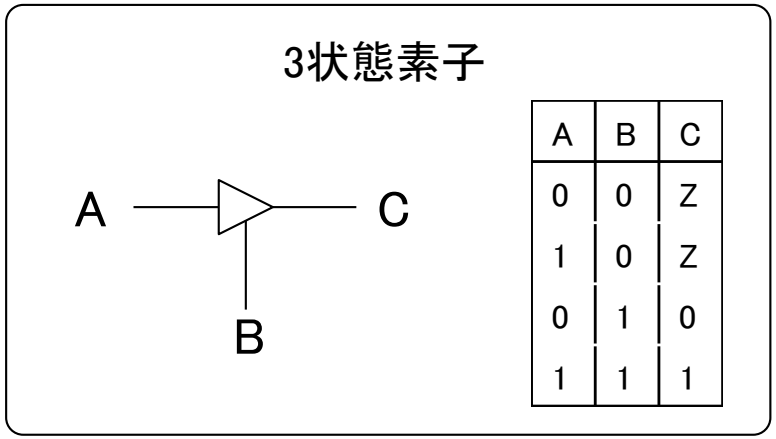
主要なレジスタの働き

- プログラムカウンタ（PC）
  - 現在実行しているプログラムの番地を格納するレジスタ
  - 参照されると自動的に内容が「＋１」される
  - プログラムの流れの制御を実現
- 命令レジスタ（IR）
  - 主メモリから読み出した命令を一時記憶
- メモリアドレスレジスタ（AR）
  - 主メモリに対してこれから操作するアドレスを格納するレジスタ
  - 主メモリ装置はここを参照して指定されたアドレスにアクセス
- メモリデータレジスタ（MR）
  - 主メモリから読み出されたデータを格納する
  - 主メモリに書き込むデータを格納する

バス

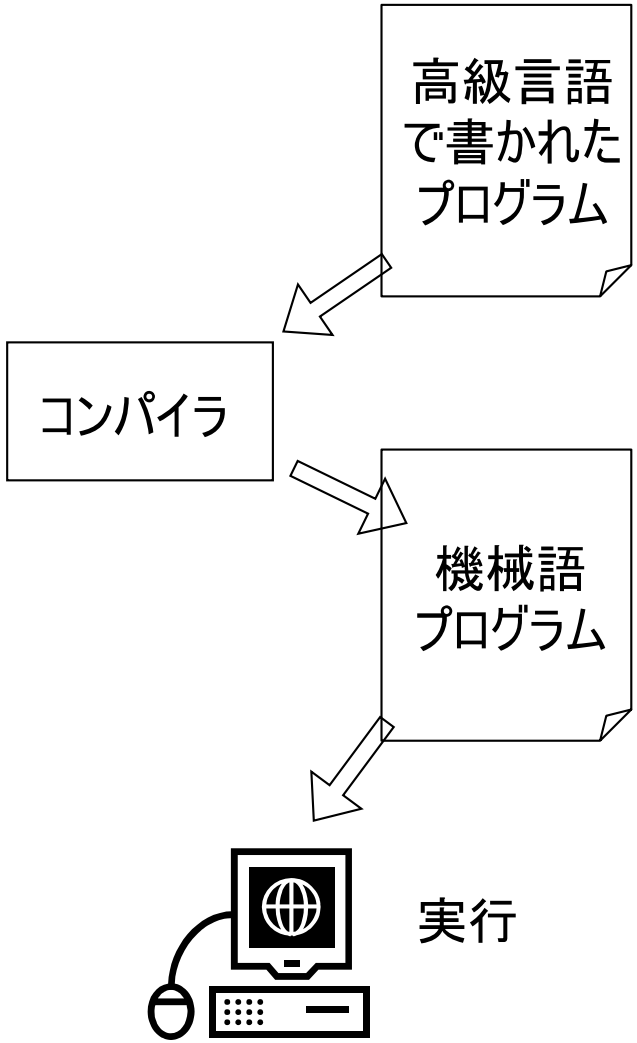
- レジスタ同士やALUとの間を共通の転送路（バス）で結ぶ
  - 転送路を少なくできる
  - 同時に2組の要素間でデータをやり取りできない
  - 3状態素子で制御を実現





プログラム

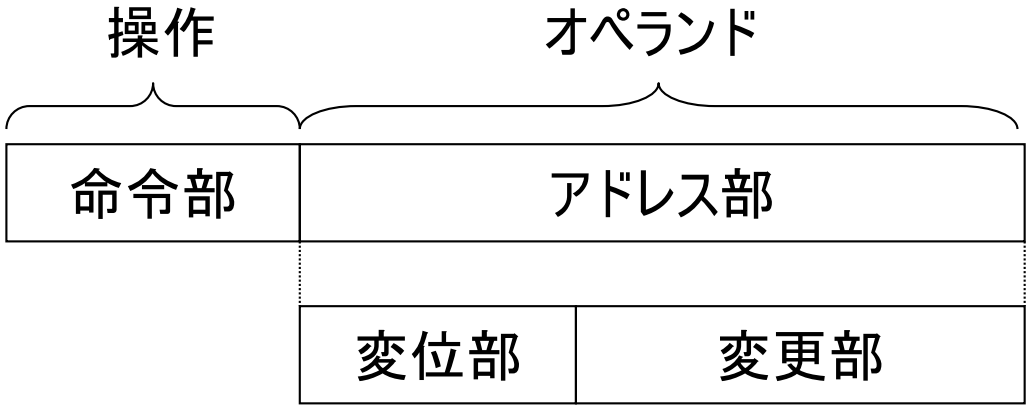
- 機械語，アセンブリ言語，コンパイラ



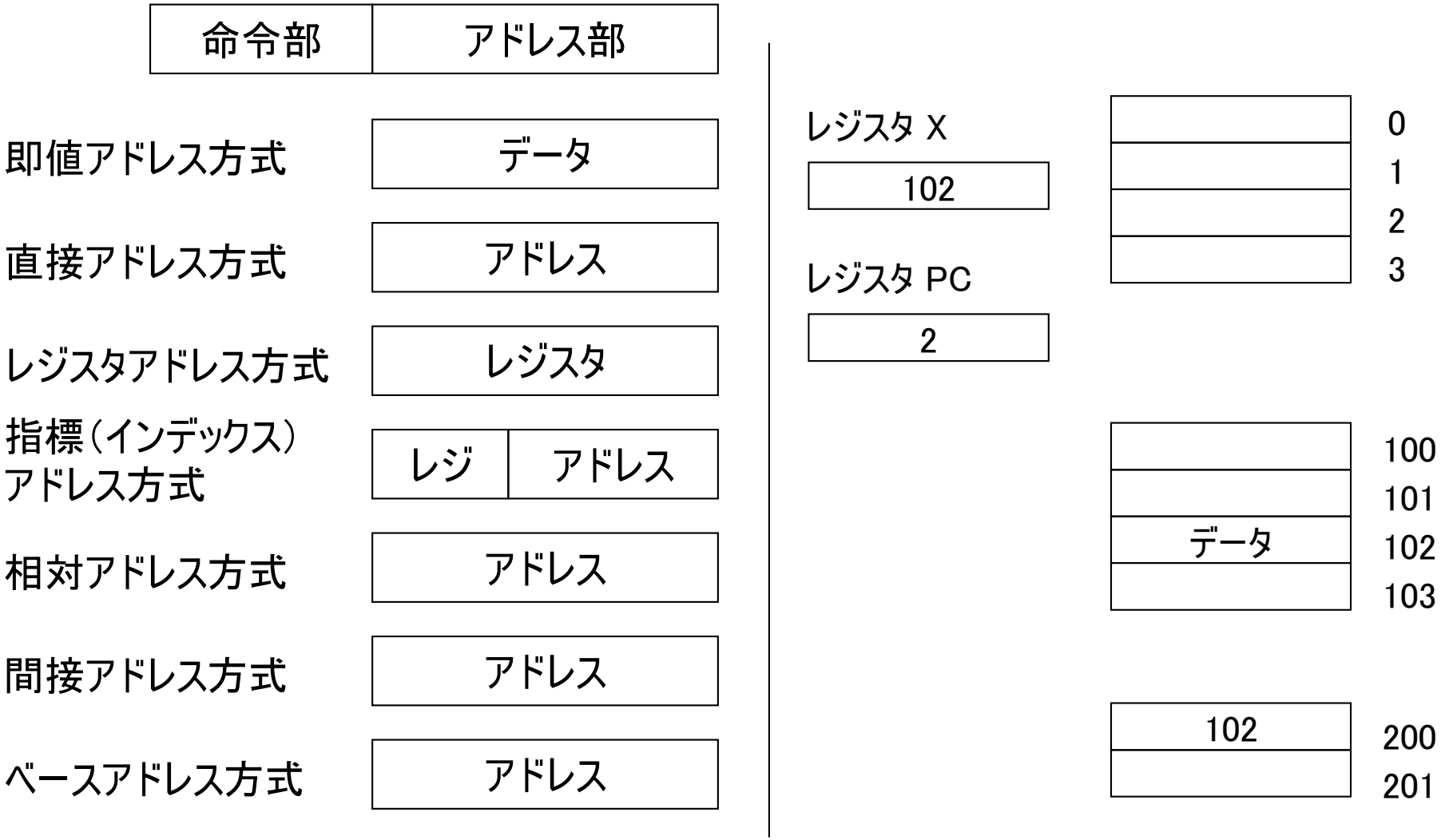
アドレス	機械語	アセンブリ言語表現	
010	0201	LD	201
011	1203	ST	203
012	2202	ADD	202
013	0203	LD	203
014	5100	JZ	100
015	1205	ST	205
	⋮		

命令の構成

- 命令＝「操作」と「操作対象となるデータ（オペランド=被演算数）」からなる
  - 例）“LD 100” = 操作(LD) + オペランド(100)
- ひとつの命令を1語～数語で表現する
- アドレス部はさらに変位部と変更部に分かれる
  - 複数のアドレス部をもつ方式もある（1～4アドレス方式）

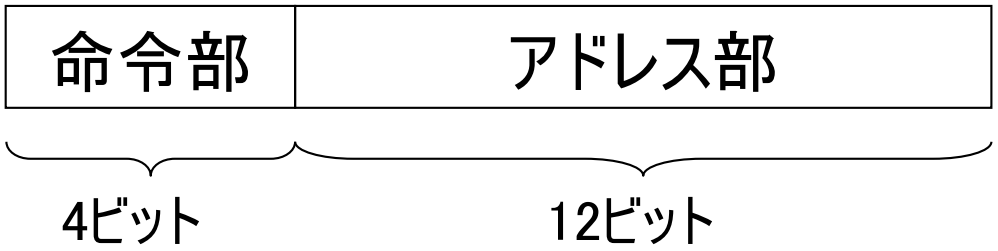


アドレス修飾


命令部
アドレス部
即値アドレス方式
データ
直接アドレス方式
アドレス
レジスタアドレス方式
レジスタ
指標(インデックス)
アドレス方式
レジ
アドレス
相対アドレス方式
アドレス
間接アドレス方式
アドレス
ベースアドレス方式
アドレス
レジスタ X
102
レジスタ PC
2
0
1
2
3
100
101
データ
102
103
102
200
201

CPU例（本講義で使うモデル）

- 1語16ビット
- アドレス空間12ビット
  - 16進では，000番地～FFF番地
- CPUの型：アキュムレータ型
- アドレス方式：直接アドレス方式
- 命令の語長：1命令＝1語（一語命令形式）
  - 命令部4ビット，アドレス部12ビット



命令セット

アセンブリ表現		機械語	意味
LD	***	0***	アドレス***のデータをAに転送
ST	***	1***	Aの内容をアドレス***に転送
ADD	***	2***	Aの内容とアドレス***の内容を加算しAに格納．演算の結果が0ならばSRを1にセット，そうでなければSRを0にする．
SUB	***	3***	Aの内容からアドレス***の内容を減算しAに格納．演算の結果が0ならばSRを1にセット，でなければSRを0に．
JZ	***	5***	SRが1ならば***にジャンプ
JMP	***	6***	アドレス***に無条件ジャンプ
SHR		C000	Aの内容を右へ1ビットシフト

機械語プログラム

例）各メモリの内容について
(201) + (202) − (204) → (203)
の演算を行い，結果が0なら100番地へ飛ぶ

アドレス	機械語	アセンブリ		意味
010	0201	LD	201	(201)→[A]
011	2202	ADD	202	[A]+(202)→[A]
012	3204	SUB	204	[A]−(204)→[A]
013	1203	ST	203	[A]→(203)
014	5100	JZ	100	もしSR=1なら100へ
015	…	…		…

プログラムの実行制御

- 主メモリに格納されたプログラムの命令を1つずつ順に取り出し(fetch)，実行(execute)する
  - 命令をメモリから取り出して実行を完了するまでの計算機の動作の一連を「命令サイクル」という
  - 命令サイクルは「取り出し(fetch)サイクル」と「実行サイクル」からなる

取り出しサイクル

1. [PC]→[AR]
2. メモリに読み出し指令（メモリからデータが[MR]に読み出される）
3. [MR]→[IR]
4. [PC]+1→[PC]
5. [IR] の上位4ビットをデコード

実行サイクル(1)

LD(転送):指定したメモリの内容をアキュムレータに

1. [IR] の下位12ビット →[AR]
2. メモリに読み出し指令(メモリからデータが[MR]に読み出される)
3. [MR]→[A] (メモリからアキュームレータへ転送)

ST(転送):アキュムレータの内容を指定したメモリに

1. [IR] の下位12ビット →[AR]
2. [A] → [MR]
3. メモリに書き込み指令 (メモリに [MR] の中身が書き込まれる)

実行サイクル(2)

ADD(加算):アキュームレータの内容とメモリの内容を加算し,結果をアキュームレータに格納

1. [IR] の下位12ビット →[AR]
2. メモリに読み出し指令(メモリからデータが[MR]に読み出される)
3. [A] + [MR]→[A] とし,結果が0ならば[SR]を1に,そうでなければ0にセット

JZ(条件分岐):状態レジスタ(SR)が1のときのみ指定のアドレスにジャンプ

1. [SR] が 1なら [IR] の下位12ビット →[PC]