

2019. 5.13

電子計算機工学

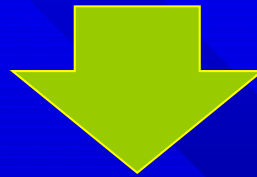
Ibaraki Univ. Dept of Electrical & Electronic Eng.

Keiichi MIYAJIMA

基本命令セット アーキテクチャ

基本アーキテクチャ

- コンピュータシステムの論理的構造を示す言葉



コンピュータのハードウェアやソフトウェア構成の考え方や、プロセッサの命令セット (instruction set) 等を指す

基本アーキテクチャ

コンピュータのハードウェアやソフトウェア構成の考え方や、プロセッサの命令セット (instruction set)等を指す



マシン命令 (machine instruction)の集合

コンピュータアーキテクチャ

命令セットアーキテクチャ

(instruction set architecture : ISA)

デジタル情報

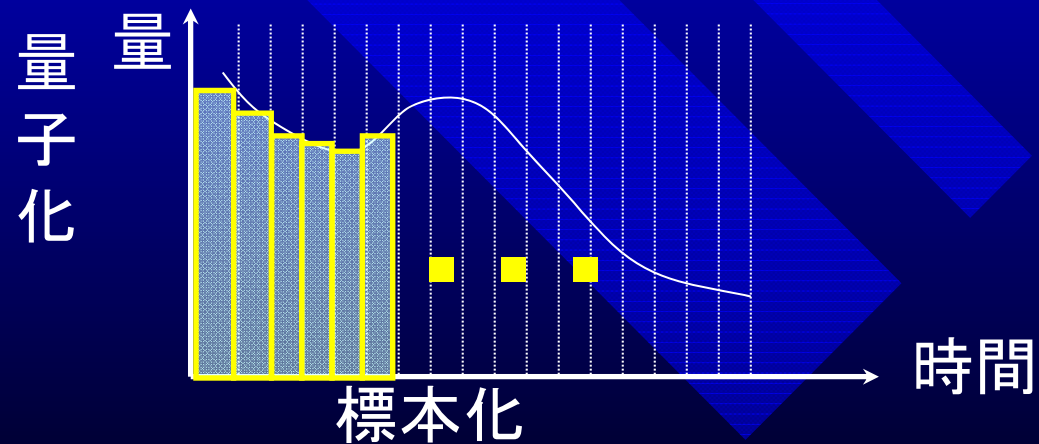
デジタル情報

コンピュータで扱える形式の情報

アナログ情報

時間経過とともに変化する情報

コンピュータで扱うために標本化を行い
デジタル情報に変換する。



コンピュータ内部での容量の単位

コンピュータ内部では0と1の2進数(バイナリ(binary))しか取り扱えない。

ビット bit 2進一桁

バイト byte 1バイト=8bit

ワード word 語 通常は4バイト(32ビット)を意味する。

各コンピュータで扱えるデータの容量をさす場合もある。

プロセッサアーキテクチャー

命令語 (プログラム)

マシン語(マシン命令)

データ

計算や処理の対象となる情報

命令とデータ

命令 (プログラム)

マシン語(マシン命令)

今実行している命令のアドレスを格納しておく場所: **プログラムカウンタ**

データ

計算や処理の対象となる情報

データのメインメモリアドレスを記憶する装置: **メモリアドレスレジスタ(MAR)**

命令形式

命令語の構成要素

- ・命令コード 命令の種類を示す
- ・オペランド データの格納場所などを示す情報



4バイト(32ビット)長命令形式が多いが、
大型計算機では、8バイト長(64ビット)もある。

命令とデータ

命令語 (プログラム)

マシン語(マシン命令)

データ

計算や処理の対象となる情報

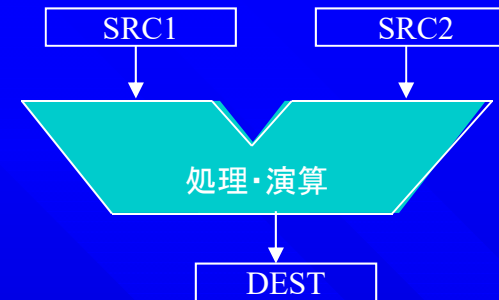
オペランド

オペランド

- ・ソースオペランド
(ソースアドレス)
- ・デスチネーションオペランド
(デスチネーションアドレス)

対象となる格納場所

- ・メインメモリ
- ・レジスタ



命令例

ADD A, B

・AレジスタとBレジスタの値を加算してAレジスタへ格納せよ

命令コード

オペランド

オペランド

オペランド数による命令形式の分類

(1) 3アドレス形式

命令コード	オペランド	オペランド	オペランド
-------	-------	-------	-------

ADD A, B, C

- ・AレジスタとBレジスタの値を加算してCレジスタへ格納せよ
(R-R-R形式の例)

(2) 2アドレス形式

命令コード	オペランド	オペランド
-------	-------	-------

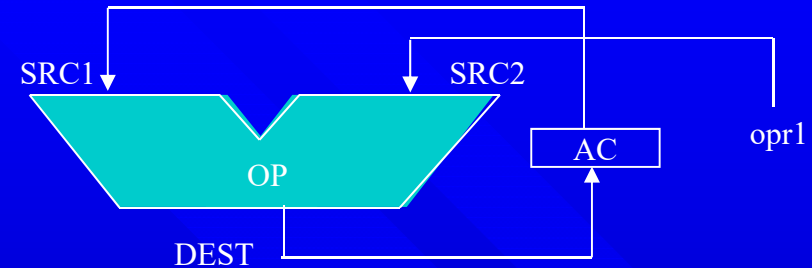
ADD A, B

- ・AレジスタとBレジスタの値を加算してAレジスタへ格納せよ
(R-R形式の例)

オペランド数による命令形式の分類

(3) 1アドレス形式

命令コード	オペランド
-------	-------



ADD B

- ・アキュムレータの値とBレジスタの値を加算してアキュムレータへ格納せよ

(4) 1・1／2アドレス形式

命令コード	オペランド	オペランド
-------	-------	-------

ADD A, メモリアドレス等

- ・AレジスタとBレジスタの値を加算してAレジスタへ格納せよ
(R-M形式の例)

オペランドの一方は必ずレジスタを指定すること

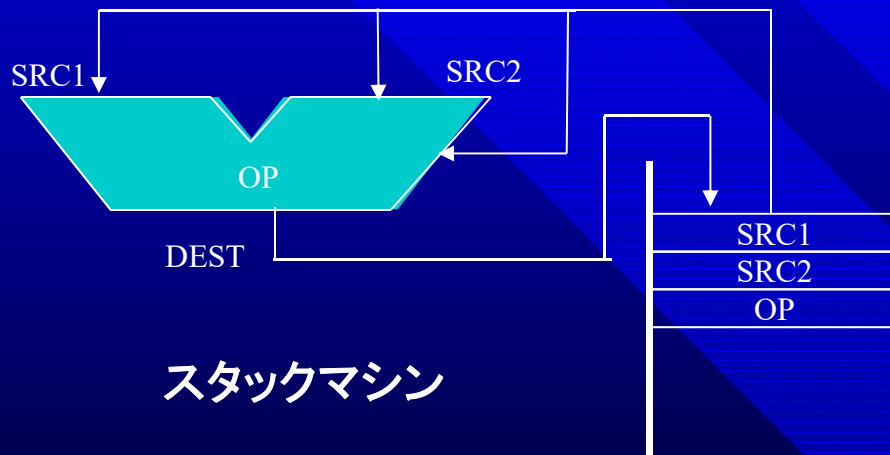
オペランド数による命令形式の分類

(5) 0アドレス形式

命令コード

オペランド

オペランドを“スタック”という一方向のみからアクセスする特殊な格納装置として暗黙指定しておくことで、オペランドを明示する必要がない命令形式



固定長命令と可変長命令

(1) 命令長固定・オペランド数固定

命令でコードステージの高速化
順序制御用ハードウェアの簡素化

(2) 命令長可変・オペランド数固定

ハードウェアコストをある程度おさえながら命令種類の多種多様化が図れる

(3) 命令長可変・オペランド数可変

多種多様な高機能命令を実現するための柔軟性を確保できる



コスト高

アドレス指定方式

アドレス指定方式(addressing): 命令の対象となるオペランドをどのように指定するか

絶対アドレス指定方式(absolute addressing):

メモリのアドレスを指定

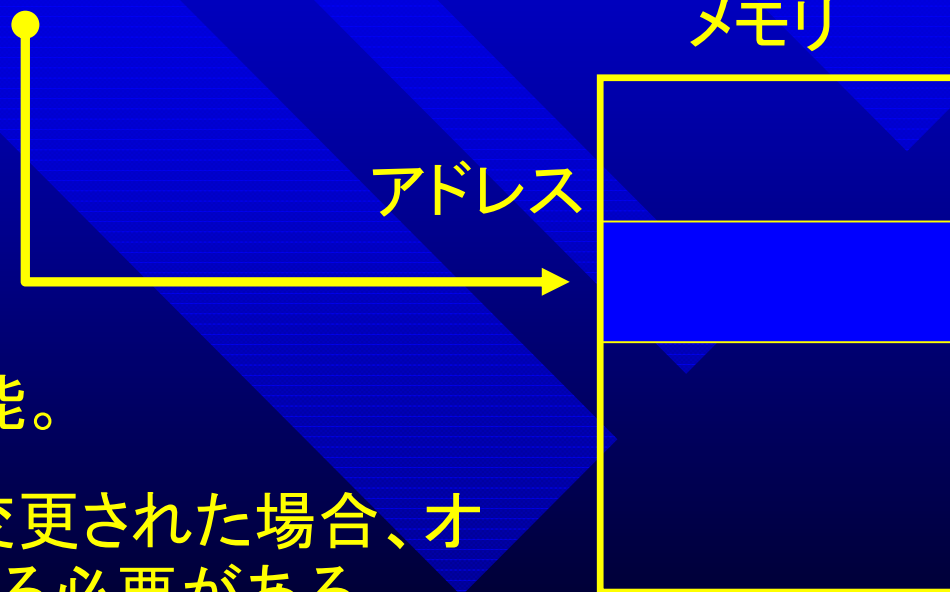
- 直接アドレス指定
- 間接アドレス指定
- レジスタアドレス指定
- レジスタ間接アドレス指定

アドレス指定方式

直接アドレス指定方式(direct addressing):

メモリのアドレス(実効アドレス:命令が実際にアクセスするためのアドレス)を直接指定

命令コード オペランド1



高速なアクセスが可能。

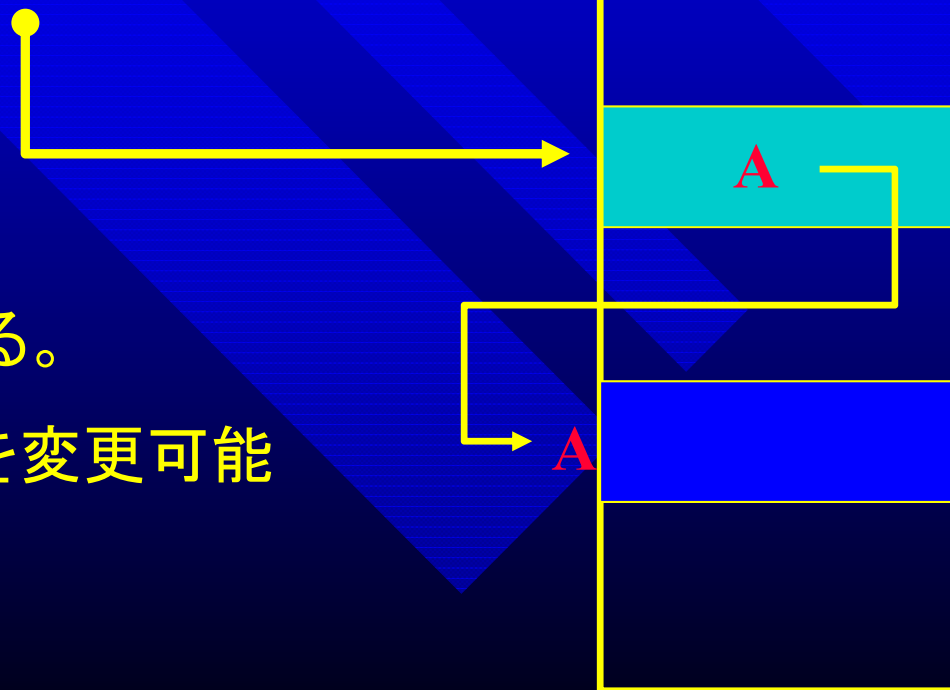
オペランドの位置が変更された場合、オペランド部を書き換える必要がある。

アドレス指定方式

間接アドレス指定方式(indirect addressing):

オペランドの値が指定したメモリのアドレスの内容が、実効アドレスとなるアドレス指定

命令コード オペランド1



アクセス時間が延びる。

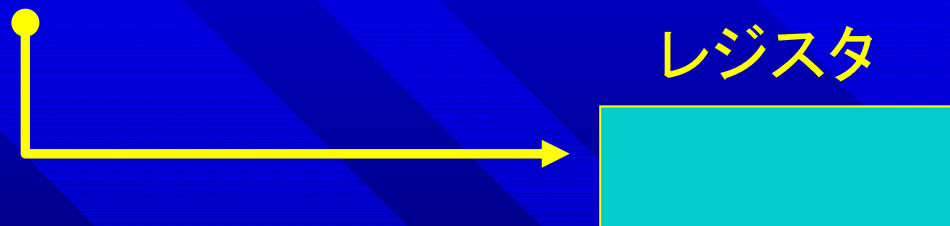
容易にアドレス指定を変更可能

アドレス指定方式

レジスタアドレス指定方式(register addressing):

直接アドレス指定方式において、アクセスの対象がレジスタの場合

命令コード オペランド1



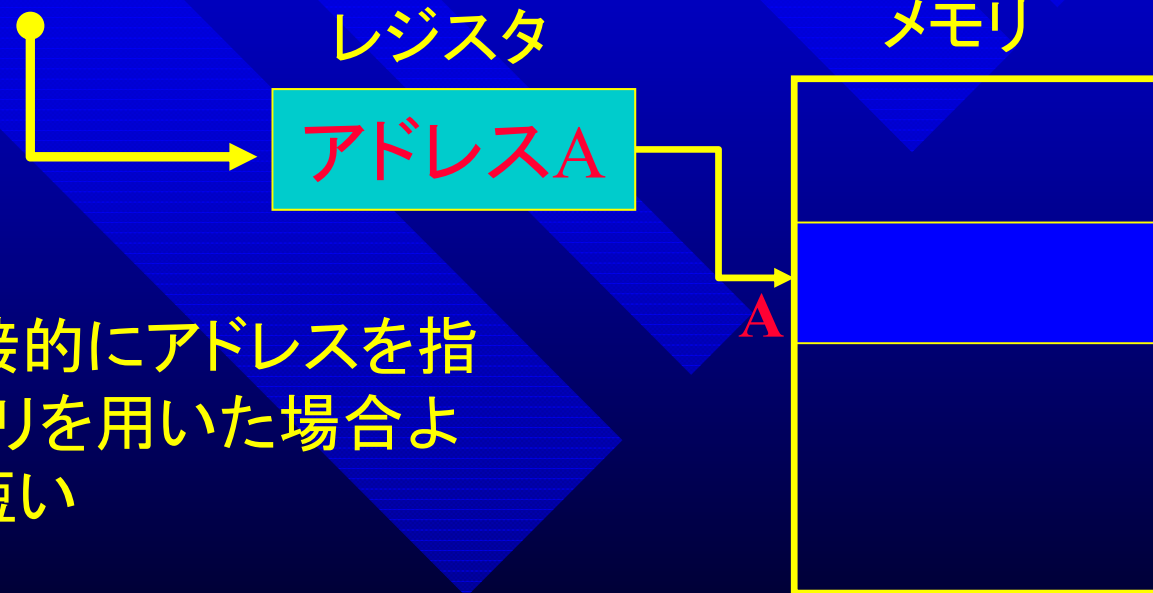
高速なアクセスが可能

アドレス指定方式

レジスタ間接アドレス指定方式:

間接アドレス指定方式において、実効アドレスがメモリではなくレジスタに存在する場合

命令コード オペランド1



レジスタを用いて間接的にアドレスを指定しているので、メモリを用いた場合よりもアクセス時間が短い

アドレス指定方式

相対アドレス指定方式(relative addressing):

アクセス対象のメモリの先頭アドレス:ベースアドレス(base address)

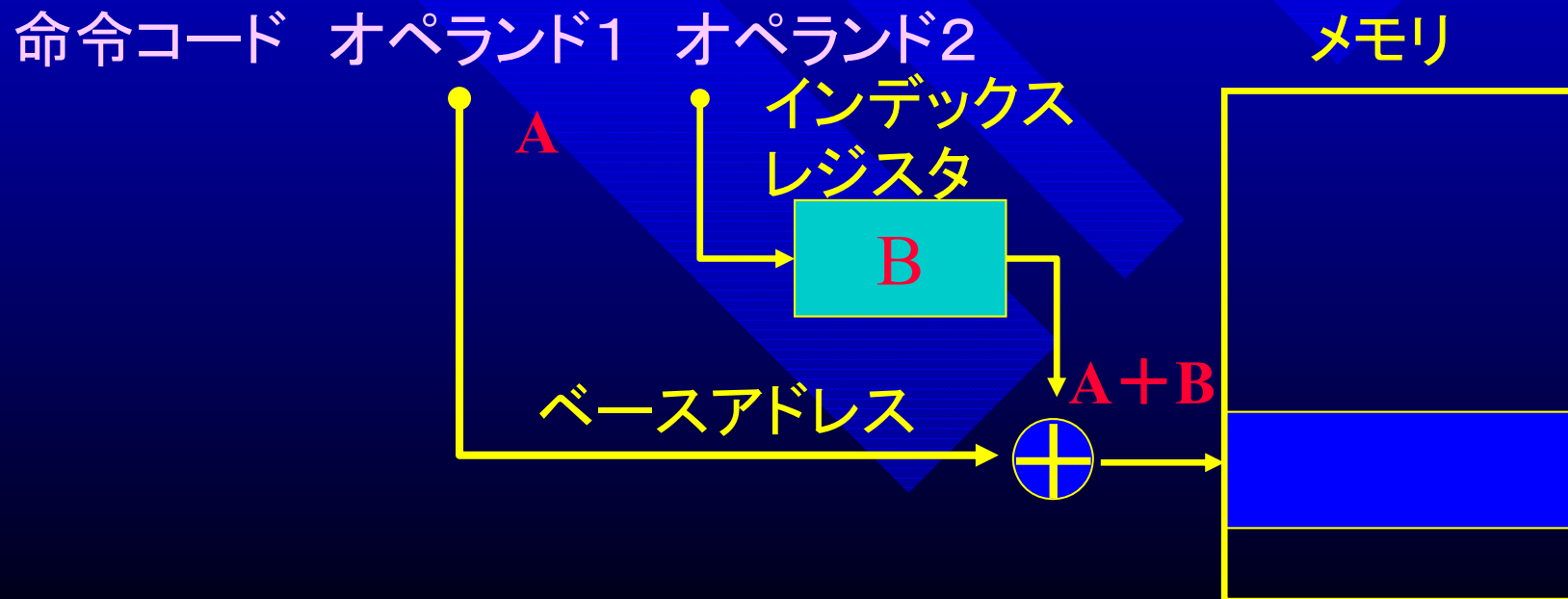
ベースアドレスとベースアドレスからの距離のようなもの(displacement)を加算して、実行アドレスとなるアドレスの指定を行う

- インデックスアドレス指定
- ベースアドレス指定
- ベースインデックスアドレス指定
- 相対間接アドレス指定

アドレス指定方式

インデックスアドレス指定方式(index addressing):

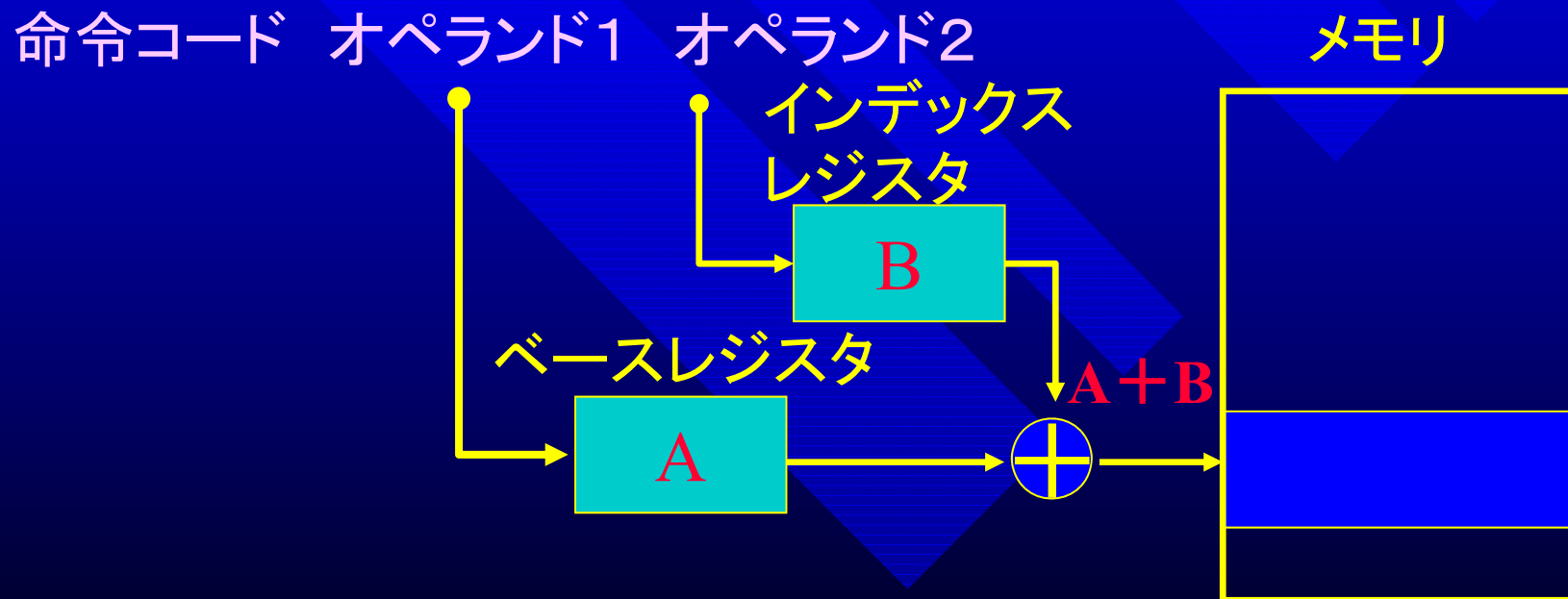
命令のオペランドでベースアドレスとインデックスレジスタ(index register)を指定する方式。ベースアドレスとインデックスレジスタの値が加算されてメモリの実行アドレスが決まる



アドレス指定方式

ベースインデックスアドレス指定方式(base index addressing):

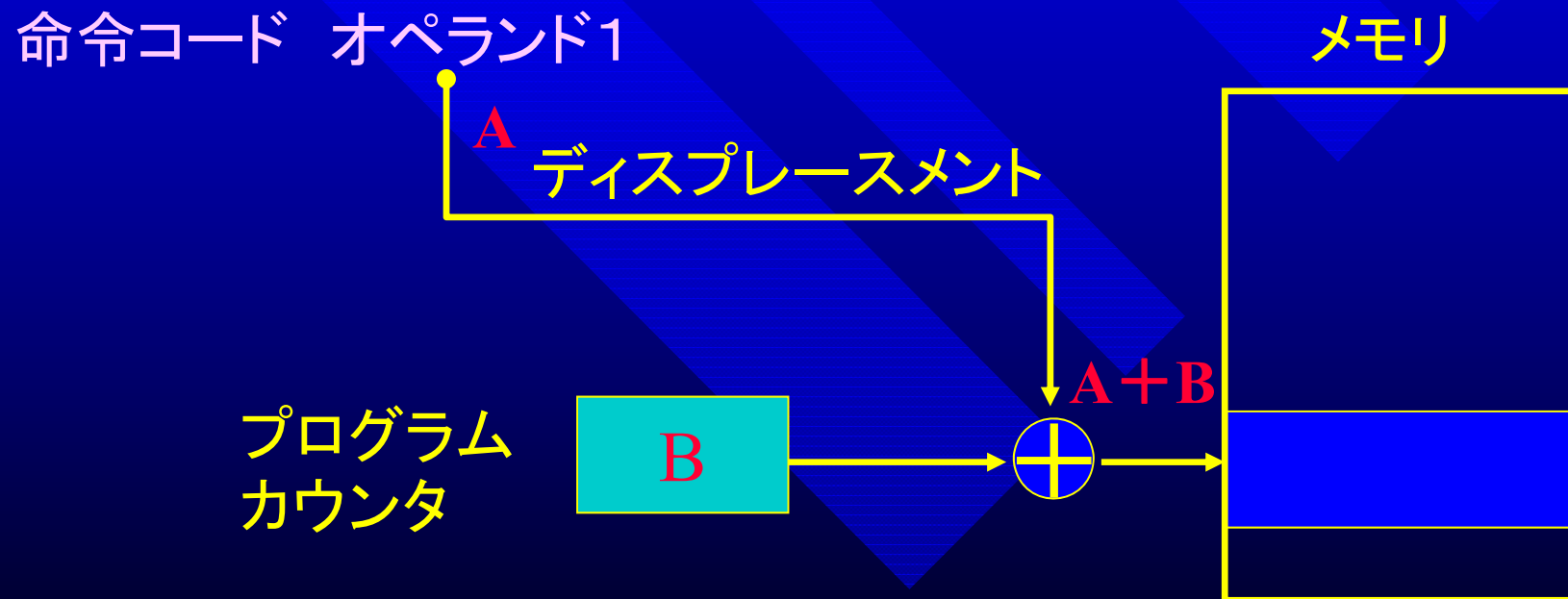
命令のオペランドでベースレジスタとインデックスレジスタを指定する



アドレス指定方式

相対アドレス指定方式(relative addressing):

命令のオペランドでディスプレースメントが与えられ、プログラムカウンタとの加算が実効アドレス



アドレス指定方式

イミディエイトアドレス指定方式(immediate addressing):

命令のオペランド値そのものがアクセス対象のデータ

命令コード オペランド



- レジスタに定数を格納する。
- メモリからオペランドを読み出す必要がないので高速に実行

データ形式については後日、述べる。

命令サイクル

プロセッサへの命令はどのように実行されるか？

命令の読み出しから演算実行までの一連の処理の繰り返시를命令サイクルという。

- ・命令の取り出し(フェッチfetch)
プログラムカウンタの更新
- ・命令のデコード
命令形式が複雑な場合この部分(ステージ)が長くなる
- ・オペランド(処理データ)の取り出し(フェッチfetch)
- ・演算実行
- ・結果の格納

基本的には上記の繰り返し

- ・次命令アドレス決定
プログラムの分岐があるときに使用する

命令セット

プロセッサの性能に関わるもっとも基本的なアーキテクチャ

データ転送命令: レジスタからレジスタへデータを転送したり、レジスタとメモリ間のデータを制御する命令

- (a) メモリからレジスタへ ロード(load)命令
- (b) レジスタからメモリへ ストア(store)命令
- (c) 転送元と転送先を入れ替える 交換(exchange)命令
- (d) スタックにデータを入れる プッシュ(push)命令
- (e) スタックからデータを取り出す ポップ(pop)命令

命令セット

演算命令:

(a) 算術演算命令

加減乗除、大小比較、データを+1するインクリメント (increment)、-1するデクリメント (decrement) 命令など

(b) 論理演算命令

論理積、論理和、否定、排他的論理和などの論理演算がビットごとに行われる

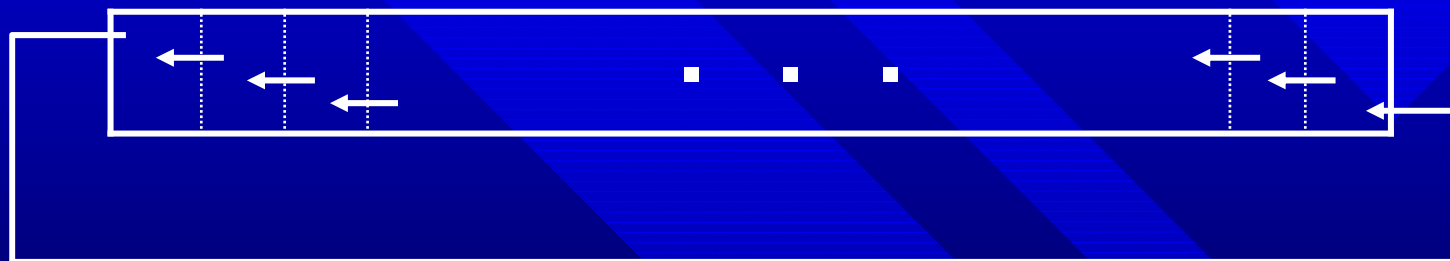
命令セット

演算命令:

(c) ビット列操作命令

フォーマット変換命令

算術シフト、論理シフト、循環シフトの各命令がある



循環シフト命令の例

命令セット

演算命令:

(c) ビット列操作命令

コード変換(code conversion)命令

コンピュータ内部で表現されるコード形式を変換、
2進数と10進数との間の基数を変換する命令

ビット操作(bit manipulation)命令

データの任意のビットを1にするセット(set)、0にするリセット(reset)、反転(complement)、0か1かを判定するテスト(test)など

命令セット

プログラム制御命令：プログラムの実行順序を制御する命令

(a) 無条件分岐(unconditional branch)命令

無条件でオペランドをアドレス変換して、プログラムカウンタに設定

(b) 条件付き分岐(conditional branch)命令

演算の結果、例えば大小関係をキャリフラグ(carry flag)やゼロフラグ(zero flag)を調べることによって判定し、その結果によってプログラムカウンタへ設定するアドレスを変更する。

条件が成立する場合と不成立の場合とで、分岐先が異なる。

命令セット

プログラム制御命令:

(c) サブルーチンコール(subroutine call)命令

プログラムの途中でサブルーチンが呼び出されて実行されると、呼び出した元のプログラムへ戻る

戻りのアドレスは、サブルーチンへの分岐命令の次の命令

戻る時はリターン(return)命令

(d) システムコール(system call)命令

実行中のユーザのプログラムから、オペレーティングシステムに制御を移す。

命令セット

システム制御命令:

入出力装置の動作の起動を開始 START I/O

入出力装置の状態を確認 TEST I/O

起動を停止 HALT I/O

プログラム状態レジスタのセット/リセット

システムの環境を操作する命令で、特に**特権命令**
(privileged instruction)とも呼ばれる

命令セット

コンディション:

実行結果の結果によるコンディション。条件分岐命令の条件として使用する。

算術演算の派生データ

オーバーフロー、桁上げ、ゼロ除算等、正負

論理演算データの結果

フラグ

コンピュータのハードウェアの状態

ハードウェアとシステムプログラム

命令セットアーキテクチャーはハードウェアとソフトウェアの機能分担の境界線

命令セットを設計することは、ハードウェア／ソフトウェアのトレードオフを設計すること。

ただし、ソフトウェア互換性を十分考慮しなければならない。

システムプログラム

プログラミング言語処理ソフト・・・ コンパイラなど

オペレーティングシステム(OS)

コンピュータアーキテクチャーを設計する作業とは、

コンパイラやOSとハードウェアのトレードオフ(機能分担)を考えること。

設計者が考慮すべき事項

- ・マシン命令形式
- ・アドレス指定モード
- ・データ型とその内部表現方式
- ・命令セットアーキテクチャー

OSとのインターフェース

- ・プロセッサ管理
- ・メモリ管理
- ・入出力処理
- ・割り込みや例外の処理
- ・実行制御

プログラミング言語の処理

プログラムとマシン語のギャップを埋める機能。

コンパイル

プログラムの実行に先立ってマシン命令列に翻訳しておくこと。

コンパイル時の最適化処理によって、ハードウェアの機能を最大限に引き出すことができる。

インタプリタ

プログラムを直接解釈して直ちに実行していく方式。

インタプリタはあらかじめシステムプログラムの一部として組み込んでおく。

実行時コンパイル

中間言語までコンパイルしておき、実行時にコンパイルしながら、
実行していく言語処理方式。

…ジャストインタイムコンパイル

変更のあった部分など、必要部分だけ実行にコンパイルする

インクリメンタルコンパイルもこの方式の一部。

RISCとCISC

プロセッサをLSI化するとき

RISC (reduce instruction set computer)

基本的な命令セットのみに限定することにより、処理回路の単純化と高速な処理を実現。

命令機能が制限されているので、プログラムのステップ数が増加。(モトローラ68000系など)

ゲーム機などはこちらが使われる

CISC (complex instruction set computer)

多数の複雑な命令も処理できる機能を有する。

回路が複雑化→コスト高・命令の解読に時間

(インテル Coreシリーズなど)

現在のパソコンはほとんどがこちら

命令機能の評価指標

プロセッサを評価するとき

TPI (平均命令実行時間)

1命令の平均実行時間(秒)

$$TPI = TPC \times CPI$$

TPC (マシンサイクル時間)

1マシンサイクル(クロック)何秒か?

CPI (平均命令実行サイクル数)

1命令を平均何マシンサイクル(クロック)で実行できるか?

RISCは意図的にCPIを小さくしている。

CISCはCPIがRISCに比べて大きくなる

本日のまとめ

基本命令セットアーキテクチャ

- **ソフトウェアの構成要素**
デジタル情報、2進数とコンピュータ内部での容量
- **命令形式**
命令、オペランド、オペランドと命令形式
- **アドレス指定方式**
- **命令実行サイクル**
- **基本命令セット**
- **ハードウェアとシステムプログラム**

本日の課題 1

1. 基本命令セットについて、それぞれの命令機能を実現するハードウェア機構の複雑さ順に並べて説明せよ。
2. CPUのアーキテクチャの視点からRISCとCISCを比較し、その特徴を述べよ。
(基本情報 改題)

本日の課題 2

3. 以下に記号語(アセンブラ言語)によるプログラムとメモリの内容を示す。(1)~(3)の各方式によるプログラム実行後のレジスタA,Bの値を示せ。ただし、数値は全て10進数とする。

LD A, 100

LD B, 200

ADD A, B

HALT

(1) イミューディエイト方式

(2) 絶対アドレス指定方式の直接アドレス指定方式

(3) 絶対アドレス指定方式の間接アドレス指定方式

番 地	内 容
100	201
101	202
102	203
:	
200	301
201	302
202	303
:	
300	411
301	412
302	413