

# コンピュータシステム基礎

情報工学科      阿部倫之

■ 担当教員： 阿部倫之

■ [abe@neptune.kanazawa-it.ac.jp](mailto:abe@neptune.kanazawa-it.ac.jp)

■ 講義資料

配布資料、eシラバス

■ オフィスアワー

野々市キャンパス：月曜 5 限 2 1 号館 4 階教員控え室

八束穂キャンパス： 6 5 号館 2 1 0 室（要予約）

# 授業運営

- 日程： 1EP1, 1EP3
  - 第1週 (9/25)
  - 第2週 (10/2)
  - 第3週 (10/16)
  - 第4週 (10/23) レポート1 出題
  - 第5週 (10/30) 小テスト1 実施、レポート1 提出
  - 第6週 (11/5(月) 5限 補講 23.221教室)
  - 休講 11/6
  - 第7週 (11/13) レポート2 出題
  - 第8週 (11/20) 小テスト2 実施、レポート2 提出
  - 第9週 (11/27)
  - 第10週 (12/4) レポート3 出題

# 授業運営

- 日程（つづき）： 2EP1, 2EP3
  - 第 1 1 週（12/11）レポート 3 提出
  - 第 1 2 週（12/18）
  - 第 1 3 週（1/8）
  - 第 1 4 週（1/15）
  - 第 1 5 週（1/22） 期末試験、 最終レポート提出
  - 第 1 6 週（1/29） 自己点検授業

## 小テスト2

- 実施日：平成30年11月20日（火） 授業時間
- 出題範囲：

### 第4章 コンピュータにおける計算の仕組み

#### 4.1 メインメモリの構成

#### 4.2 メインメモリの容量と番地

#### 4.3 番地指定（アドレス指定）

#### 4.4 メモリ内容の表示方法

#### 4.5 命令と実行

- 命令処理のサイクル
- 命令の構造
- 番地指定方式

※問題4.6

## レポート2

- 提出日：平成30年11月20日（火）授業開始前
- レポート範囲：

補足問題集 3ページ ～ 5ページ（問題0.18は除く）

注意1）片面印刷で提出

注意2）ホッチキス止め不可（そのまま重ねて提出）

注意3）全ページにクラス番号、名前を記載

# 第7週

- 小テスト1 返却
- レポート2と小テスト2について
- 第4章

## 4.5 命令と実行

4.5.1 命令とは

4.5.2 命令処理のサイクル

4.5.3 命令の種類

4.5.4 命令の構造

4.5.5 オペランドの指定方式

4.5.5 番地指定方式

## 4.5 命令と実行

### 4.5.1 命令とは

- 機械命令（**命令**）
- 機械語（マシン語、機械命令ワード）
  - 命令コード（**オペコード**、**OPコード**）とオペランドで構成
- 機械命令プログラム（ネイティブコード）
  - 命令ワードをメモリ上に並べたもの
  - 命令ワードは連続している必要がある。不連続な場合には、分岐命令を用いて**プログラムカウンタ**（**IA:命令アドレスレジスタ**）を強制的に書き換える必要がある
  - 命令を実行するとき、プログラムカウンタは次の命令ワードを参照するように自動的に更新される



## 4.5.2 命令処理のサイクル(命令サイクル)

- (1) 命令読み出し (命令フェッチ) : 命令アドレスレジスタ (プログラムカウンタともいう) に保持しているメインメモリの番地 (アドレス) から機械命令ワードを読み出して命令レジスタに格納 (一時記憶) する. (注: フェッチとは取り出す意味)
- (2) 命令デコード (解読) : 命令レジスタの機械命令ワードのコードをシーケンサで解読し, CPU内部の制御信号やデータレジスタのワード番号, データワードを格納しているメインメモリの番地 (アドレス) を生成する.
- (3) データ読み出し (オペランドフェッチ) : データレジスタ (汎用レジスタ) またはメインメモリから処理対象のデータワードをバスへ読み出す.
- (4) 演算実行 : データワードをALUに入力し, デコードした機械命令で指定している演算を実行する.
- (5) 結果の格納 : ALUから出力される演算結果のデータワードをバスを經由してデータレジスタ (汎用レジスタ) に格納する.

(6) 次の命令番地 (アドレス) の決定 : 現在実行している機械命令ワードの次に実行する機械命令ワードが格納されているメインメモリの番地 (アドレス) を決定し, 命令アドレスレジスタに格納する **次の命令ワードを参照できるようにIAを更新する**

(注: ただし (6) は, 通常の命令では (4) の開始時に実行し, (4) と (5) の期間に次の命令の(1)を平行に実行する. )

## 4.5.3 命令の種類

- 算術演算命令：加算命令，減算命令，比較命令など
- 論理演算命令：AND命令，OR命令，XOR命令，NOT（反転：インバート）命令など
- ビット列操作命令：上位／下位シフト命令，上位／下位回転シフト命令など
- データ移動命令：データレジスタ（汎用レジスタ）のワード間移動命令（注：データレジスタは通常複数ワード分の記憶容量をもつ），メインメモリからのデータ読み出し／書き込み命令など
- 分岐命令：命令アドレスレジスタ（プログラムカウンタ）に保持している次に読み出す命令のメインメモリの番地を離れた（分岐先）番地に強制的に書き換える強制分岐命令と，算術演算結果（正，負，ゼロなど）によってこれらの番地を切り替える条件分岐命令がある．
- 特殊命令：空命令（何もしない命令），一時停止命令，終了停止命令など
- その他：特定の制御用レジスタに対してデータ書き込み／読み出しを行う命令，カウンタ機能をもつ命令など

## 4.5.4 命令の構造

1つの機械命令のビット列は次の2つの部分からなっている。

- ・ オペコード（オペレーションコードの略，OPコードまたは命令コードとも書く）：加算やデータ移動などの操作（演算）の種類を指定する．また次のオペランドの形式およびそれが指すデータの形式も指定している．
- ・ オペランド（操作（演算）されるものという意味）：操作（演算）の対象となるデータの所在，結果を格納する場所などを指定する．

オペコード	オペランド
-------	-------

オペランド部に記述する内容には、

- 1) メインメモリ（主記憶）の番地
- 2) データレジスタ（汎用レジスタ）のワード番号
- 3) 即値（処理対象データそのもの）

などがある。



## ※仮想的な機械命令を使用して説明する。

オペコード	第1オペランド	第2オペランド	...
-------	---------	---------	-----

0番レジスタと1番レジスタを加算して、結果を0番レジスタに格納する命令の例（アセンブラ形式）

### ・2オペランド形式

AR   GR#0, GR#1  
オペコード   第1オペランド   第2オペランド

### ・1オペランド形式（アキュムレータ形式）

第1オペランドを0番レジスタに固定。固定されたレジスタを「アキュムレータ」と呼ぶ。

AR   GR#1  
オペコード   第2オペランド

- ・ ソース と ディスティネーション

0番レジスタ と 1番レジスタを加算して、結果を0番レジスタに格納する命令の例 (アセンブラ形式)

<u>AR</u>	<u>GR#0,</u>	<u>GR#1</u>
オペコード	第1オペランド	第2オペランド

GR#0 <- GR#0 + GR#1

第1オペランド : ソースオペランド

第2オペランド : ディスティネーション (宛先) オペランド

## 4.5.5 オペランドの指定方式

### (1) オペランド明示方式

RISC方式のプロセッサに多い形式

AR    GR#0, GR#1  
オペコード   第1オペランド   第2オペランド

### (2) オペランド暗黙指定方式

CISC方式のプロセッサに多い形式

- ・ 1 オペランド形式（アキュムレータ形式）

AR    GR#1  
オペコード   第2オペランド

## 4.5.6 番地指定方式

### ※オペランドが主記憶（メモリ）のデータを参照する場合

- ・仮想的な機械命令を使用して説明する。
- ・問題4.6（5）の事例を用いる。
- ・第1オペランドが 0番レジスタ（GR#0） に固定されたアキュムレータ形式の加算命令を用いて番地指定方式を説明する。
- ・プログラムは600番地から格納されており、説明に使用する加算命令は700番地に入っている。
- ・本来、機械語命令は16進数で表現するが、説明の都合上、アセンブラ形式で表現する。また、データやアドレスは10進数で表現する。

- ・プログラムの先頭は600番地
- ・プログラムの先頭番地は12番レジスタに入れるものとする  
(プログラムの先頭を指すレジスタをベースレジスタと呼ぶ)
- ・加算命令は700番地

IA (命令アドレスレジスタ)

700

(番地) 主記憶(プログラム)

600

機械語  
プログラム

GR#0  
(0番レジスタ)

0

GR#2  
(2番レジスタ)

500

GR#12  
(12番レジスタ)

600

700

(加算命令)

主記憶(データ)

番地(アドレス)  
内容(データ)

200	400	600	700	800	900
400	500	300	900	700	800



## 4.5.6 番地指定方式

第1オペランドが 0番レジスタ (GR#0) に固定されたアキュムレータ形式の加算命令を用いて説明する。この加算命令は、700番地に入っており、また数値は10進数で表現する。

### (1) 直接番地方式

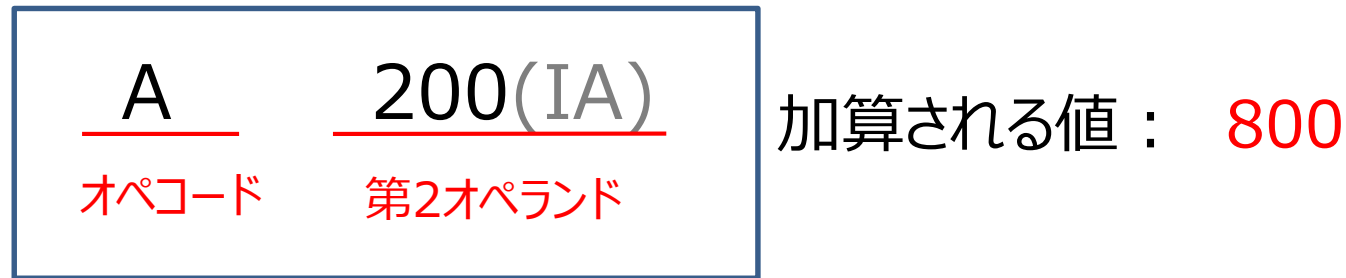
0番レジスタと メモリの200番地の内容を加算して、結果を0番レジスタに格納する命令の例。

<u>A</u>	<u>200</u>
オペコード	第2オペランド

加算される値： 400

## (2) 命令相対番地方式

0番レジスタと メモリの (IA+200) 番地の内容を加算して、結果を0番レジスタに格納する命令の例。



IA (命令アドレスレジスタ、プログラムカウンタ)

700

IA: 命令アドレスレジスタ (プログラムカウンタ)

※実行すべき命令のメモリ番地が入っている

※レジスタに加算される数値 200 を変位と呼ぶ

### (3) プログラム相対番地方式

0番レジスタと メモリの（プログラムの先頭番地+200）番地の内容を加算して、結果を0番レジスタに格納する命令の例



加算される値： 700

GR#12  
(12番レジスタ)

600

ベースレジスタ: プログラムの先頭番地が入っている。  
※汎用レジスタで代用する。

※レジスタに加算される数値 200 を変位と呼ぶ

## (4) レジスタ相対番地方式

0番レジスタと メモリの (GR#2 + 200) 番地の内容を加算して、結果を0番レジスタに格納する命令の例



加算される値： 900

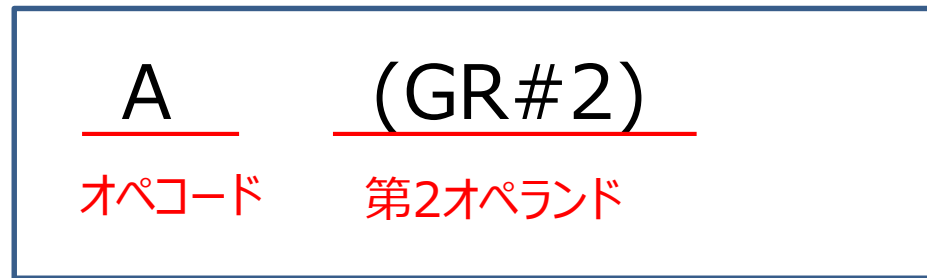
GR#2  
(2番レジスタ)

500
-----

インデックスレジスタ: 処理対象のデータの先頭番地が入っている。  
※汎用レジスタで代用する。

## (5) レジスタ間接番地方式

0番レジスタと 2番レジスタの内容が指す番地の内容を加算して  
、結果を0番レジスタに格納する命令の例



加算される値 : 600

GR#2  
(2番レジスタ)

500
-----

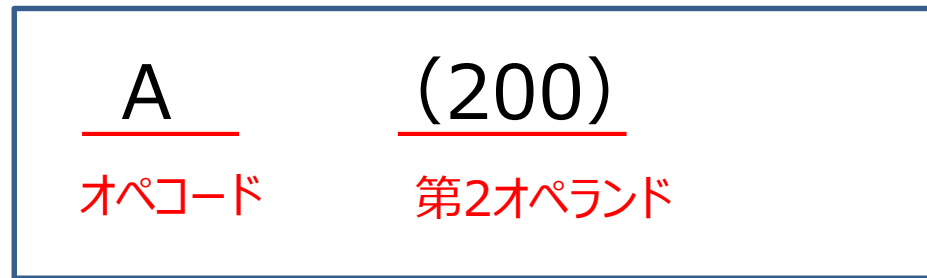
主記憶(データ) : 追加

番地(アドレス)  
内容(データ)

200	400	600	700	800	900	500
400	500	300	900	700	800	600

## (6) メモリ間接番地方式

0番レジスタと メモリの200番地の内容が指す番地の内容を加算して、結果を0番レジスタに格納する命令の例



加算される値： 500

400番地の内容を加算

主記憶(データ)

番地(アドレス)	200	400	600	700	800	900
内容(データ)	400	500	300	900	700	800

## (7) 即値方式（イミディエート方式）

0番レジスタと オペランドの内容を加算して、結果を0番レジスタに格納する命令の例

<u>AI</u>	<u>200</u>
オペコード	第2オペランド

加算される値： 200

※即値方式の場合は、オペコードが異なる。