

# 機械語命令の書き方とデータ転送命令

山本昌志\*

2004年8月23日

## 1 前回の復習と本日の学習

### 1.1 復習

- CASL II の命令には、3種類ある。

アセンブラ命令	機械語に変換するとき、アセンブラーに指示する命令。CPU が持っている命令 (機能) ではないので、特定のビットパターンに変換されない。
機械語命令	CPU が持っている命令 (機能) に対応する。CPU の命令に応じた特定のビットパターンに変換される。
マクロ命令	機械語命令を組み合わせたもの。

- 前回の授業では、次のアセンブラ命令について学習した。

START	プログラムの実行開始番地を示す。
END	プログラムの終わりを示す。最後に実行される命令ではない。
DC	処理すべきデータを定義する。
DS	計算に必要なメモリーの領域を確保する。

### 1.2 本日の学習内容

アセンブラ命令の学習が終わったので、機械語命令の説明を始める。機械語命令が終われば、マクロ命令である。他の2つの命令に比べて、機械語命令は数も多く、その動作也多岐にわたる。なんと言っても、アセンブラのプログラムの中心はこの機械語命令である。プログラムの目的であるデータの加工は、ほとんど機械語命令で実行される。

この命令は、CPU の機能を表しているので、注意深い諸君は、CPU を作るために必要な機能の概略を理解できるであろう。実際、CASL II の命令と同じものが、世の中で使われているほとんどの CPU に実装されている。実際の CPU は、CASL II より多くの命令を持って、より複雑な命令にも対応している。

---

\*国立秋田工業高等専門学校 電気工学科

機械語命令は数が多いので、しばらくは機械語命令の学習を進めることになる。命令の文法は教科書に書いてある。プリントではその補助的な説明を行うことにする。

本日の学習のゴールは、

- 機械語命令の書き方が分かる。
- データ転送命令が理解できる。
  - － LD
  - － ST
  - － LAD

である。

## 2 機械語命令の書き方

### 2.1 命令形式

教科書に書かれているように、機械語命令の書き方は、オペランドが異なる 5 種類に分類できる。機械語命令の場合、CASL II の 1 行は機械語命令の 1, 2 ワードのマシン語に変換されて、メインメモリーに格納される。1 ワードは 16 ビットで、2 ワードは 32 ビットである。その 1 あるいは 2 ワードで、命令の種類と対象であるオペランドを示すマシン語になる<sup>1</sup>。

機械語命令の書き方は、オペランドの指定の仕方により、以下の 5 通りに分類できる。

ラベル欄	命令コード欄	オペランド欄	注釈欄
[label]	OP	r1,r2	;レジスタ同士の操作
[label]	OP	r,adr[,x]	;レジスタとメモリーの操作
[label]	OP	adr[,x]	;メモリーの操作
[label]	OP	r	;レジスタの操作
[label]	OP		;メモリーやレジスタを操作しない

これを見て分かるように、機械語命令の多くは、メモリーやレジスターを操作する。プログラムの目的は、データを処理することで、そのデータはメモリーまたはレジスターに格納されることから、そのことが理解できる。

### 2.2 オペランドの内容

命令の対象となるオペランドの書き方は、先に示したように、5 種類である。機械語命令の書き方を見よ。これらの内、r1 と r2, r が汎用レジスタを示している。GR0 とか、GR1 と書く。汎用レジスタの範囲は、GR0 から GR7 までである。

x は指標レジスタを示している。指標レジスタについては、第 6 回の授業で説明したが、忘れていると思うので、再度説明する。プログラムを書いているとき、基準点のアドレスにある値を加算してデータにアク

<sup>1</sup>マシン語への変換は、教科書の p.213 を見よ

セスしなくてはならないことがある。このようなときに指標レジスタを使う。CASL II では、オオペランド欄に、

```
adr, x
```

と書けば、それが出来る。adr が基準点のアドレスで、x が指標レジスタである。実際にデータが操作される実行アドレスは、adr+x ということになる。adr は、つぎに述べる方法でアドレスを指定する。加算する値を格納するのは指標レジスタ (index register) で汎用レジスタの GR1 ~ GR7 を使う。どうして、GR0 はダメなのか?。命令をマシン語に直すとこの理由が分かる。このように、指標レジスタを用いて、アドレスを操作することをアドレス修飾と言う。

## 2.3 アドレス

メインメモリーの中に格納されているプログラム (命令とデータ) にアクセスしないと、CPU は何もできない。メモリーの特定の場所の内容を参照するために、記憶領域に応じて番地が振り分けられている。その番地のことをアドレスと言う。

COMET II のメインメモリーのアドレスは、16 ビットである。従って、アドレスの範囲は、0 ~ 65535 (#0000 ~ #FFFF) 番地となる。このメモリー空間は、20 年くらい前の 8 ビットパソコンと同じである。64k バイトです。ちなみに、いま主流の 32 ビットパソコンのメモリー空間は 32 ビットで、4G バイトにもなる。メモリーにアクセスする場合、番地を指定しなくてはならない。その番地の指定方法を述べます。

教科書に書かれている通り、アドレスは、つぎに示す 3 通りの方法で記述する。最初の 2 つの 10 進数と 16 進数を使う場合、絶対アドレスを指定することになる。よっぽどのことがないかぎり、絶対番地を指定することはない<sup>2</sup>。なぜならば、実際のプログラムを実行する場合、データがどの番地に格納されているかは、プログラムは分からない。プログラム実行段階で、OS が決めるからである。従って、みなさんは最後のアドレス定数を使うことになる。

10 進定数	10 進数の定数を用います。内容は、教科書に書かれている通りです。
16 進定数	16 進数の定数を用います。16 進数であることを表すために、先頭に # を付けます。
アドレス定数	ラベル名を指定します。アセンブラーにより、ラベル名がアドレスに変換されます。

## 2.4 リテラル

機械語命令のオペランドの *adr* は、アドレスを示すことは先に述べた通りである。アドレスの指定は、10 進定数と 16 進定数、アドレス定数がある。さらに、リテラルでもそれを指定できる。リテラルを用いたアドレスは、10 進定数や 16 進定数、あるいは文字定数の前に '=' の記号をつける。

教科のリテラル形式をアセンブルすると、DC 命令を使ったのマシン語になる。あとは、教科書の通り。

<sup>2</sup>実際は、メモリーの番地ではないが、数値を使うことがある。後で説明する。

### 3 データ転送命令

メインメモリからレジスタに、あるいはレジスタからメモリにデータを転送する命令の使い方を示す。いずれの場合も、1回のデータの転送量は1ワード(16ビット)である。メモリやレジスタの領域は複数あるので、その位置を指定しなくてはならない。メモリーの場合は先ほど示した方法でアドレスを指定する。レジスタの場合はその名前で転送場所を指定する。

#### 3.1 レジスタへの転送(LD)

##### 3.1.1 内容

役割 レジスタにデータを転送する。

LD:Load

書式	ラベル欄	命令コード欄	オペランド欄
	label	LD	r1,r2
label	LD	r,adr[,x]	

- フラグレジスタの値は変化する。

COMET IIでは、算術演算や論理演算は必ず汎用レジスタ上で行われる<sup>3</sup>。そのため、演算の対象となるデータを汎用レジスタに格納しなくてはならない。LD命令は、メインメモリーのあるアドレスのデータを汎用レジスタにコピーする命令として使われる。そればかりではなく、汎用レジスタ間のコピーにも使われる。しかし、汎用レジスタからメインメモリーや、メインメモリーからメインメモリーへのコピーはできない。メインメモリーへのコピーはST命令を使う。

語源は、英語のloadです。loadのにはいろいろの意味がありますが、その中で'読み込む'と意味で使われています。パソコンでアプリケーションを実行するとき、ハードディスクにあるプログラムをメインメモリーへ読み込むこともロードといいます。ネットでのファイルの受け渡しのことも、ダウンロードやアップロードと言う。

##### 3.1.2 例

```
LD GR0,GR1 ;GR1の内容をGR0へコピー
LD GR0,A ;ラベルAが示す番地の内容をGR0へコピー
LD GR0,A,GR1 ;(A+GR1)番地の内容をGR0へコピー
LD GR1,GR1 ;GR1の符号をチェック
```

<sup>3</sup>演算はCPUが行うため、その記憶領域であるレジスタが使われるのは当たり前である。

## 3.2 メモリーへの転送 (ST)

### 3.2.1 内容

役割 レジスタの内容をメモリーへ転送する。

ST:STore

#### 書式

ラベル欄	命令コード欄	オペランド欄
label	ST	r, adr [, x]

- フラグレジスタの値は変化しない。

ST 命令は、LD 命令とは逆に、汎用レジスタの内容をメインメモリの指定番地にコピーする。語源は STore で、' 備蓄する ' などの意味がある。書式や機能は教科書に書かれている通りである。

### 3.2.2 例

```
ST  GR0,A      ;GR0の内容をメインメモリのA番地へコピー  
ST  GR0,A,GR1 ;GR0の内容を(A+GR1)番地へコピー
```

## 3.3 アドレスの転送 (LAD)

### 3.3.1 内容

役割 実効アドレスを汎用レジスタにロードする。

LAD:Load Address

#### 書式

ラベル欄	命令コード欄	オペランド欄
label	LAD	r, adr [, x]

- フラグレジスタの値は変化しない。

LD 命令は、メインメモリの指定した番地の内容 (データ) を汎用レジスタにコピーする。一方、LAD 命令は、その番地 (実効アドレス) を汎用レジスタにコピーする。

実効アドレス (adr,[x]) の指定に 10 進定数や 16 進定数を指定することにより、直接、汎用レジスタに値を格納することができまる。そのため、汎用レジスタに初期値を設定したり、制御変数の値を操作するときに使われ、非常に使い勝手のある命令になっている。

### 3.3.2 例

```
LAD  GR0,A      ; ラベル A の実効アドレスを GR0 へコピー
LAD  GR0,A,GR1  ; (A+GR1) を GR0 コピー
LAD  GR0,4      ; GR0 の内容を 4 に設定
LAD  GR1,0,GR2  ; GR2 の内容を GR1 へコピー
LAD  GR1,3,GR2  ; (3+GR2 の内容) を GR1 へコピー
LAD  GR1,1,GR1  ; GR1 の値を 1 増加させる
LAD  GR1,-1,GR1 ; GR1 の値を 1 減少させる
```