機械語命令の書き方とデータ転送命令

山本昌志*

2005 年 12 月 16 日

1　ここでの学習

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，その動作も多岐にわたる．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，COMET II と同じような命令が，世の中で使われているほとんどの CPU に実装されている．市販の CPU は COMET II より複雑な命令にも対応しており，プログラマーには便利になっていいる．

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．

ここでのの学習のゴールは，

- 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` レジスターへのデータ転送

  `ST` メモリーへのデータ転送

  `LAD` レジスターへアドレス転送

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

*国立秋田工業高等専門学校　電気工学科

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[1]．

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も，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．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，オペランド欄に，

```
adr,x
```

と書く．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る．

[1] マシン語への変換は，教科書の p.213 を見よ

教科書に書かれている通り (p.39)，アドレスは，つぎに示す 3 通りの方法で記述でる．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[2]．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，プログラマは分からない．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 進数の定数を用いる．内容は，教科書に書かれている通り．
16 進定数	16 進数の定数を用いる．16 進数であることを表すために，先頭に#を付ける．
アドレス定数	ラベル名を指定する．アセンブラーにより，ラベル名がアドレスに変換される．

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 (p.39).

3 データ転送命令

メインメモリーからレジスタに，あるいはレジスタからメモリー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．メモリー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．レジスターの場合はその名前で転送場所を指定する．

3.1 レジスターへの転送 (LD)

3.1.1 内容

役割 レジスター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]

[2]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．

- フラグレジスターの値は変化する.

COMET II では，算術演算や論理演算は必ず汎用レジスター上で行われる[3]．そのため，演算の対象となるデータを汎用レジスター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，汎用レジスター間のコピーにも使われる．しかし，汎用レジスター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 1oad である．load のにはいろいろ意味があるが，その中で' 読み込む' と意味で使われている．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う．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いう.

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
```

3.2 メモリーへの転送 (ST)

3.2.1 内容

役割 レジスター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]

- フラグレジスターの値は変化しない.

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 (p.42).

3.2.2 例

```
ST  GR0,A      ;GR0 の内容をメインメモリーの 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]演算は CPU が行うため，その記憶領域であるレジスターが使われるのは当たり前である.

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ーにロードする.

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]

- フラグレジスターの値は変化しない.

LD 命令は，メインメモリー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
```