

マクロ命令

山本昌志*

2005年1月31日

1 学習内容

これまでの講義で、アセンブラ命令と機械語命令の学習を終えた。最後に残ったマクロ命令をここで学習する(教科書 [1] の p.83~p.86)。その前に、CASL II の命令について、再度、見直しておく。

アセンブラ命令 教科書では、非実行文と書かれているものである。アセンブラという変換プログラムに対して、いろいろな指示を行う命令である。プログラム実行時には、COMET II の CPU の動作の指示は行う命令ではない。したがって、この命令は機械語に変換されて特定のビットパターン (1 と 0 の組み合わせ) に変換されることはない。

START	プログラムの先頭を定義
	プログラムの実行開始番地を定義
	他のプログラムで参照する入口名を定義
END	プログラムの終わりを明示
DC	定数を定義
DS	領域を確保

ただし、DC 命令は、それに引き続く値にビットパターンに変換される。DS 命令はビットパターンに変換されないが、必要な領域を確保する。この 2 つは、FORTRAN の変数宣言と同じような働きをする。実際のプログラムでは、データの値を定義することに使われる。

機械語命令 COMET II の CPU に動作の指示を行う。そのため、この命令に対応した論理回路が、CPU の中に組み込まれている。これら命令は、アセンブラにより特定のビットパターンの機械語に変換され、そのパターンに従い、論理回路が動作する。実行時には、そのビットパターンが主記憶装置に格納されている。

* 国立秋田工業高等専門学校 電気工学科

LD, ST, LAD	データの移動
ADDA, SUBA, ADDL, SUBL	加算・減算演算
AND, OR, XOR	論理演算
CPA, CPL	比較演算
SLA, SRA, SLL, SRL	シフト演算
JPL, JMI, JNZ, JZE, JOV, JUMP	分岐処理
PUSH, POP	スタック操作
CALL, RET	サブルーチンの呼び出しと戻り
SVC, NOP	その他

マクロ命令 マクロ命令とは、特定の機能を果たす、いくつかの機械語命令の集まりに名前を付けたものである。この名前を指定するだけで、これらの命令の集まりが実行できる。これにより、頻繁に使われる定形的な命令群をマクロ命令にすることにより、同じようなプログラムをいちいち書くことを省くことができ、便利である。多くの命令から構成されるため、アセンブラーにより変換されるビットパターンは非常に多くなる。

IN	入力装置 (キーボード) から、文字データを読み込む
OUT	出力装置 (ディスプレイ) に、文字データを書き込む
RPUSH	汎用レジスタの内容を、GR1, GR2, …, GR7 の順でスタックに格納
RPOP	スタックの内容を GR7, GR6, …, GR1 の順で汎用レジスタに格納

2 マクロ命令

2.1 入出力関係

通常、入出力関係を伴ったしよりににはキーボードやディスプレイのハードウェアの制御が必要である。COMET II でそれらを制御するとなると、非常にプログラムが難しくなる¹。そこで、CASL II にはマクロという形でそれを実現している。マクロも機械語命令の集まりであるが、プログラマーはそんなことを気にしないで、通常の命令と同じように使うことができる。

2.1.1 入力命令 (IN)

CASL II では、データの入力方法は 2 通りある。以前学習した DC とここで示す IN である。DC ではデータはプログラム中に記述し、アセンブル時にビットに変換され、実行時にメモリーにロードされる。それに対して、IN は実行時にキーボードから入力されたデータをメモリーに格納する。

内容

¹簡単にする方法もあるが、ここではそこまで踏み込まない。ハードウェアが関わる問題となる

命令語	IN
語源	INput (return:入力)
役割	入力装置 (通常はキーボード) から文字列を読み込む .
書式	[ラベル] IN ラベル 1, ラベル 2
機能	入力装置から 1 レコード (256 語) 分のデータが , ラベル 1 で指定された入力領域に入力される . 読み込まれた文字数はラベル 2 の領域にセットされる . ファイルの終わりを検出した場合 (データがない場合) , ラベル 2 には-1 がセットされる .
フラグレジスタ	アセンブラーに依存 .

使用例

IN TEXT,NUM ; キーボードからデータを読み込む

この命令では , 必ず読み込んだデータと文字数を格納する主記憶の領域を確保しなくてはならない . すなわち , 以下が必要である .

```
TEXT DS 256 ; データ領域 , 1 レコード 予約する
NUM DS 1 ; 読み込み文字数を入れる
```

データ領域として , 1 ワード以外の値を指定した場合 , エラーになるか否かはアセンブラーに依存するであろう . シミュレーター WCASL-II ではエラーにはならなかった .

2.1.2 出力命令 (OUT)

内容

命令語	OUT
語源	OUTput (return:出力)
役割	出力装置 (通常はディスプレイ) に文字列を書き出す .
書式	[ラベル] OUT ラベル 1, ラベル 2
機能	データをラベル 2 で指定された文字数出力装置に書き出す .
フラグレジスタ	アセンブラーに依存 .

使用例

OUT TEXT,NUM ; ディスプレイに文字を書き出す

この命令では , メモリーに格納されているデータは文字として取り扱われる . 数値を表示させるときには工夫が必要である . それについては , 以降の学習範囲である .

2.2 レジスター関係

レジスターのデータの待避と復元には、機械語命令の PUSH と POP を使うことができる。これは、レジスターひとつずつを処理するため、頻繁に多くのレジスターを取り扱う場合、プログラムが長くなり不便である。そこで、汎用レジスターの GR1 ~ GR7 まで、一度の待避と復元ができるマクロ命令が用意されている。

2.2.1 レジスターの待避 (RPUSH)

内容

命令語	RPUSH
語源	???
役割	GR1, GR2, GR3, ..., GR7 の順でスタック領域にデータをプッシュする。
書式	[ラベル] RPUSH
機能	スタックポインター (SP) の値を減らしながら、汎用レジスターの値をプッシュする。
フラグレジスタ	アセンブラーに依存。

使用例

RPUSH ; レジスターのデータの待避

2.2.2 レジスターの復元 (RPOP)

内容

命令語	RPOP
語源	???
役割	スタック領域のデータをポップし、GR7, GR6, GR5, ..., GR1 の順でレジスターに入れる。
書式	[ラベル] RPOP
機能	スタックポインター (SP) の値を増やししながら、汎用レジスターの値をポップする。
フラグレジスタ	アセンブラーに依存。

使用例

RPOP ; レジスターのデータの復元

参考文献

- [1] 東田幸樹, 山本芳人, 広瀬啓雄. アセンブラ言語 CASL II. 工学図書 (株), 2002 年.