

本節の授業のテーマ

本日の授業のテーマは、以下のとおりです。

- (1) CASL II の命令の種類
- (2) プログラム例 (加算演算)
- (3) プログラムのハンドアセンブル

本日の授業のゴールは、以下のとおり。

- CASL II の命令の種類が理解できる。そして、アセンブラーやマシン語との関係がわかる。
- 加算プログラムの内容が理解できる
- プログラムのハンドアセンブルができる。

1. CASL II の命令の種類

CASL II の命令は、3 種類あります。プログラムを書く場合、これらをどのように使うか理解する必要があります。

命令の具体的な内容については、今後の学習です。ここでは、簡単に概要を述べます。

アセンブラ命令

教科書の P.28～P.35 で説明している非実行文とかかかれているものです。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す。COMET II の CPU の動作の指示は行いません。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ありません。

CASL II には、次の 4 個のアセンブラ命令があります。

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	領域を確保
DS	定数を定義

機械語命令

教科書の P.40～P.82 で説明しています。この命令は、COMET II の CCPU の動作の指示を行います。機械語命令は、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ています。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ます。ビットパターンへの変換は、教科書の P.213 に書かれています。

CASL II には、以下の 28 個のアセンブラ命令があります (教科書の P.203)。

ロード、ストア	LD ST LAD
算術、論理演算	ADDA ADDL SUBA SUBL AND OR XOR
比較演算	CPA CPL
シフト演算	SLA SRA SLL SRL
分岐	JPL JMI JNZ JZE JOV JUMP
スタック操作	PUSH POP
コール、リターン	CALL RET
その他	SVC NOP

マクロ命令

教科書の P.83～P.86 で説明しています。マクロ命令とは、特定の機能を果たすためのいくつかの命令の集まりに名前を付けたものです。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されます。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ます。

多くの命令から構成されるため、アセンブラにより変換されるビットパターンは非常に多くなります。

CASL II には、以下の 4 個のマクロ命令があります (教科書の P.203)。

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	GR 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 GR に格納

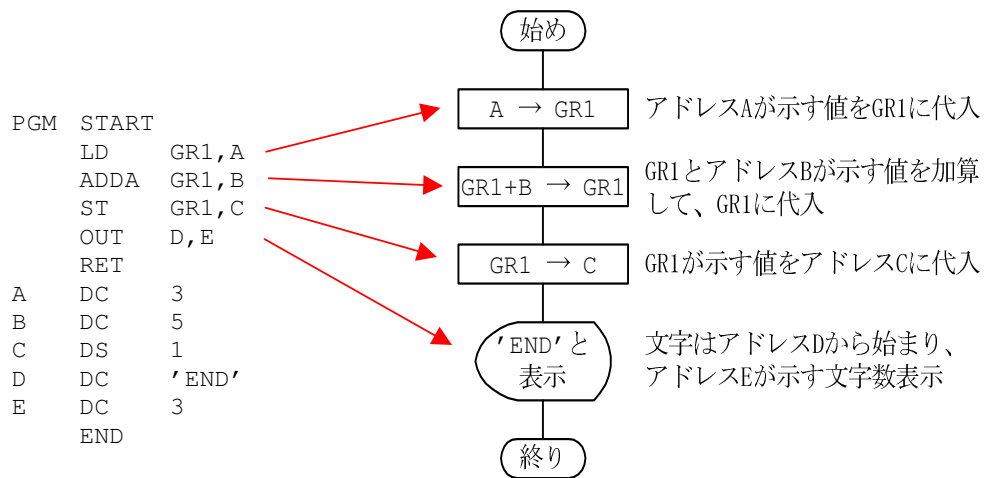
2. プログラム例 (加算演算)

それでは、実際に CASL II の命令とそれが、どのようにメモリーに格納されるか見てみましょう。そのためには、実際のプログラムを見るほうが良いでしょう。

次のような CASL II のプログラムを考えましょう。

- 加算をするプログラムです。
- 加算をする 3 と 5 をメモリーに格納しておきます。
- それを引き出して、加算を行います。
- 加算の結果を、メモリーに戻します。
- そして、最後に "END" と出力装置 で表示します。

CASL II のプログラムは、次のようになります。合わせて、フローチャートも示しますので、内容を理解してください。



実行文ではないため、フローチャートには書きませんでした。命令コードの DC と DS には、

A	DS	3	アドレス A に数値の 3 を格納
D	DC	'END'	アドレス D に文字 'E' を格納。続いて、アドレス D+1 に文字 'N' を格納。アドレス D+2 に文字 'D' を格納。
C	DS	1	アドレス C を先頭に、1 ワード分のメモリー領域を確保

の意味があります。これは、アセンブラがメモリーの中身を決めたり、確保したりします。

それでは、この CASL II がアセンブラでどのように機械語に変換されるか見てみましょう。シミュレーター WCASL-II で変換すると、図 1 のようなマシン語になりました。以下のことを理解してください。

- アセンブラ命令は、マシン語に変換されていない。
- 機械語命令は、1 対 1 の対応でマシン語に変換される。
対応については、次の章で学習します。
- マクロ命令は、複数のマシン語に変換される。
変換されるマシン語は、OS やアセンブラに依存します。したがって、マクロ命令のマシン語への変換については、ここでの学習の範囲外です。この変換を考えるためには、OS とアセンブラの設計が必要です。

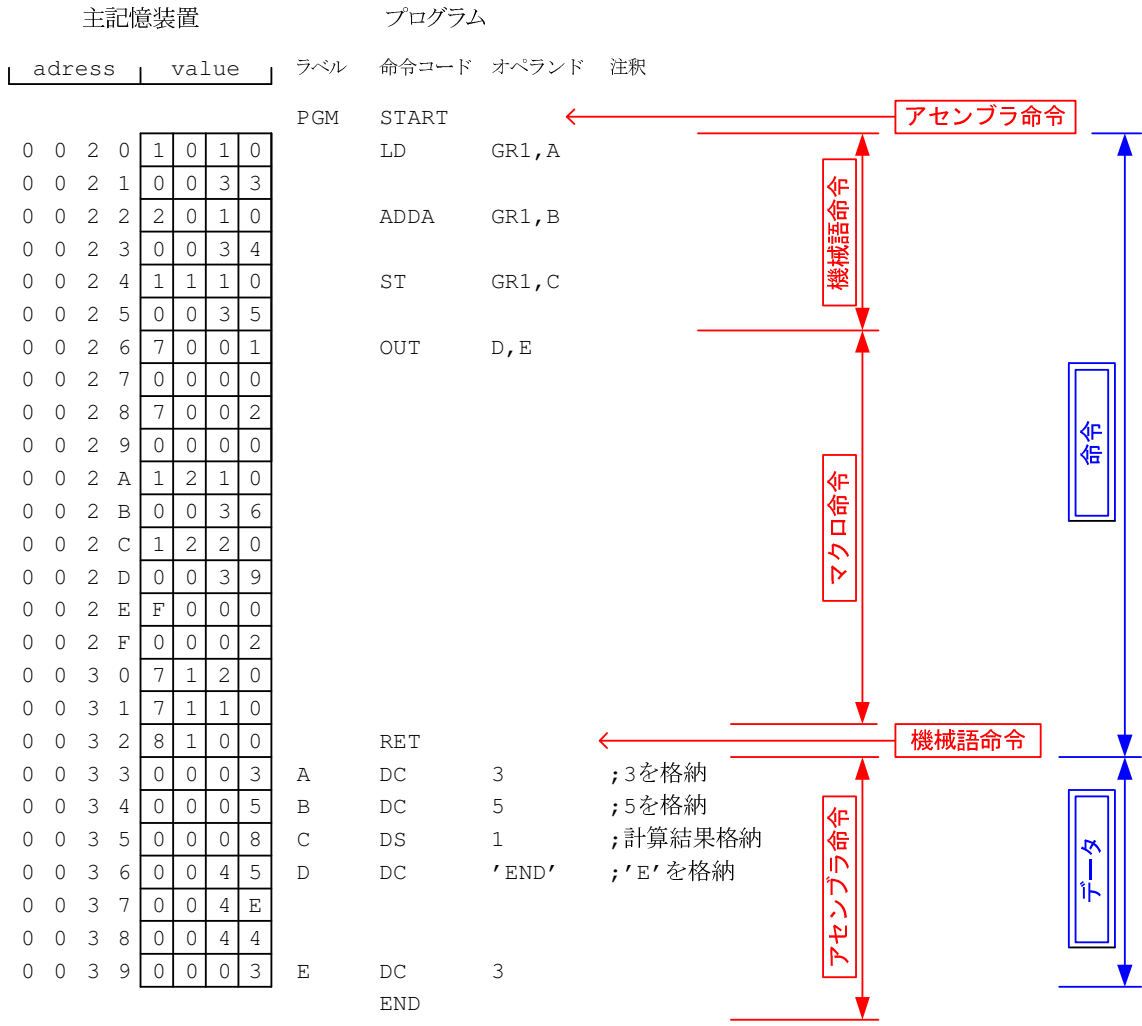


図 1 プログラムとメモリーの内容。プログラムの実行結果を示しているため、アドレス 0035 には計算結果の 8 が格納されている。

3. ハンドアセンブル(機械語命令)

機械語命令は、アセンブラ言語の仕様が分かれば、人間がアセンブルすることができます。人間がアセンブルすることをハンドアセンブルと呼びます(もしかして死語かも)。

CASL II は、教科書の P.213 を見れば、ハンドアセンブルが可能です。それでは、先ほどの命令をハンドアセンブルしてみましょう。まずは、

```
LD      GR1,A
```

です。教科書を見ると、LD は 2 ワードから構成されることが分かります。2 ワードの最初のワードは、

- ・第 12～15 の 4 ビット (1)₁₆
- ・第 8～11 の 4 ビット (0)₁₆
- ・第 4～7 の 4 ビット レジスターを示す。GR1 は、(1)₁₆
- ・第 0～3 の 4 ビット 指標レジスターを示す。無いので、(0)₁₆

です。これで、最初のワードが (1010)₁₆ となることが分かったと思います。次のワードは LD で読み込むべきアドレスです。A のアドレスの値は、OS とアセンブラーにより決められます。ここ(wcasl-II)では、(0033)₁₆ と決められました。これで、最初の命令がマシン語に変換されました。すなわち、(1010)₁₆ と (0033)₁₆ です。実行時に、これが連続した主記憶領域に格納されます。格納されるアドレスは、OS によって決められます。ここでは、アドレス (0020)₁₆ から格納されています。

あとは、同じなので、各自ハンドアセンブルしてみてください。課題として、レポートにしても良いですか?。平常点として、加点したいと思います。

課題

- ・残りの

```
ADDA   GR1,B
ST     GR1,C
```

をハンドアセンブルしなさい。結果は分かっているので、その過程を書くこと。

- ・教科書 P.17 の図 2.4 は間違っています。間違いを見つけて、訂正しなさい。

期限

- ・7月14日(月) AM 8:30