

CASL IIのプログラム例(その4)

山本昌志*

2005年1月14日

1 復習と本日の内容

1.1 復習

これまで、教科書の第5章の「CASL IIプログラム例」で学習した内容は、以下の通りである。冬休み前にこれだけのことを学習したことを思い出して欲しい。

算術演算・論理演算 算術演算はただの加算と減算、論理演算はビット毎の論理和と積と排他的論理和

条件分岐 比較とジャンプ命令により、実行する文を選択する

マスク処理 特定のビットの内容を調べる

アドレス修飾 ラベルとインデックスレジスタでアクセスするメモリのアドレスを決める

シフト演算 上手に使える、かけ算やわり算ができる

1.2 本日の内容

高級言語のプログラムは、

順次 プログラムの命令は上から下へ実行される。

選択 制御式により実行される分が選択される。これは、FORTRAN や C 言語の if 文のことである。

繰り返し ループあるいは反復とも呼ばれ、同じ命令を繰り返す。FORTRAN では DO 文、C 言語では for 文などである。

の基本構造からなる。順次についてはいままで、気が付かないで使っていたもので、なにも考えることはない。選択に関しては、例題 [3] で示した条件分岐 (比較+ジャンプ) である。本日の例題 [4] では、繰り返し文を学習する。

長いプログラムを、この基本構造だけでは作成することは不可能である。技術的には可能であるが、何が書いてあるか全く分からないプログラムになってしまい。メンテナンスが不可能である。そのため、プログ

*国立秋田工業高等専門学校 電気工学科

ラムを機能毎に細かく分割して、分かりやすくする方法がとられる。この機能毎に分割されたプログラムをサブルーチンという。FORTRAN では、SUBROUTINE とか FUNCTION というものがそれに当たる。本日の 2 番目のテーマは、このサブルーチンを CASL II で実装する方法をである。

2 [例題 6] 繰り返し処理

教科書の List5-6 のプログラムを例にして、繰り返し処理 (ループ) について説明する。

2.1 教科書の例

教科書のプログラムは、

- ラベル DATA から、ラベル KOSUU が示す語数の整数のデータが格納されている。
- このデータの最大値を探し出し、それをラベル MAX が示す領域に格納する。

という問題を解く、プログラムである。このプログラムは、教科書の p.98 の List5-6 に示されている。まず最初に、

- このプログラムの命令とデータの領域の区別

を考える。これは、さんざんやったので理解できているものとする。

このプログラムの核となる部分は、最大値を探すアルゴリズムである。教科書の例では、それは、次のようなアルゴリズムとなっている。

1. 最初に読み込むデータ (アドレス [data]) を暫定最大値とする。
2. それ以降は繰り返し処理。
 - (a) 次のデータと最大値を比較する
 - (b) 個数分のデータの比較が済んでいなければ、2a(次データ処理)に戻る

である。最大値を探すアルゴリズムには、次々にデータを最大値と比較する処理が必要である。ここに繰り返し処理が使われる。

2.2 繰り返し処理

高級言語では繰り返し専用の命令が用意されているが、アセンブラ言語にはない。そのため、条件分岐を使い繰り返しを行うことにする。アセンブラ言語では、以前学習したように、条件分岐は比較命令 (CPA, CPL) とジャンプ命令 (JMI, JNZ, JZE, JUMP, JPL, JOV) を上手にを使って、繰り返し処理を行うことになる。フローチャートで書くと、図 1 の様な構造である。

このような繰り返し構造を実現するためには、一度実行した命令に戻る必要がある。そのために、フローチャートの上へ分岐 (ジャンプ) するのである。このままだと、無限ループに陥るので、そこから抜けるた

めの機構も必要である。パラメーターの値に従いループを続けるか、そこから抜けるかを定める。それは、分岐 (比較とジャンプ命令) で実現できる。

今回の問題であれば、データの個数分だけ繰り返せばよい。そのために、カウンターを用いて、データ数のカウントをしている。これはまた、指標レジスターにも使える。

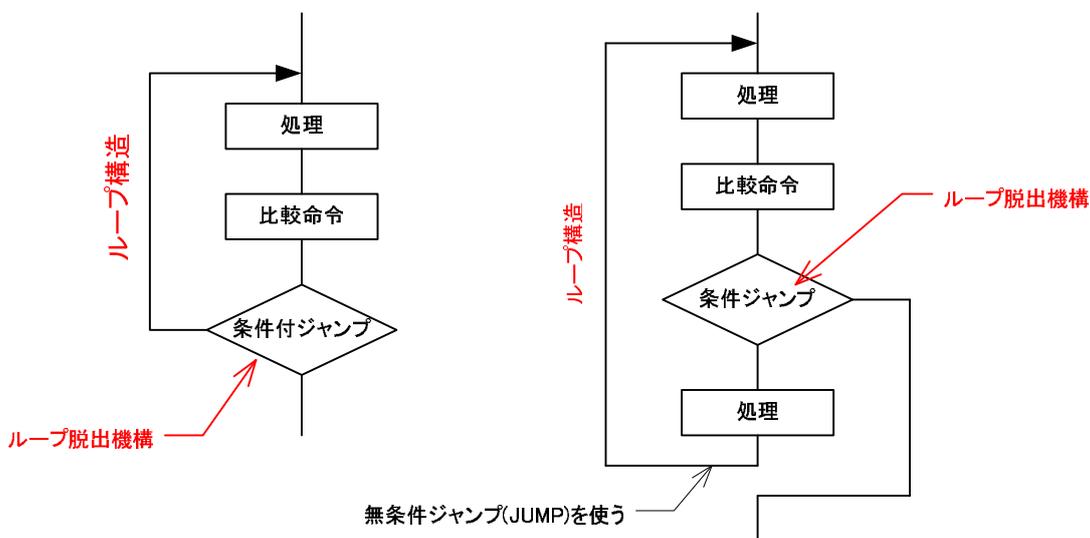


図 1: 分岐命令を使った繰り返し構造

2.3 プログラムの構造とフローチャート

このプログラムのフローチャートを図 2 に示す。ループ構造になっているのが分かるだろう。それについての説明の前に、データを取り扱うレジスターやラベルの内容を表 1 に示しておく。

表 1: 汎用レジスターとメモリの内容

GR0	読み込んだデータ (比較すべき対象) を入れる。
GR1	データ数から 1 引いた値。指標レジスタの最大値。
GR2	データのカウンタ。0 から始まり、指標レジスタとしてつかう。
DATA	調べるデータの先頭アドレス。
KOSUU	調べるデータ数が書かれているアドレス。
MAX	調べたデータの最大値を入れるアドレス。

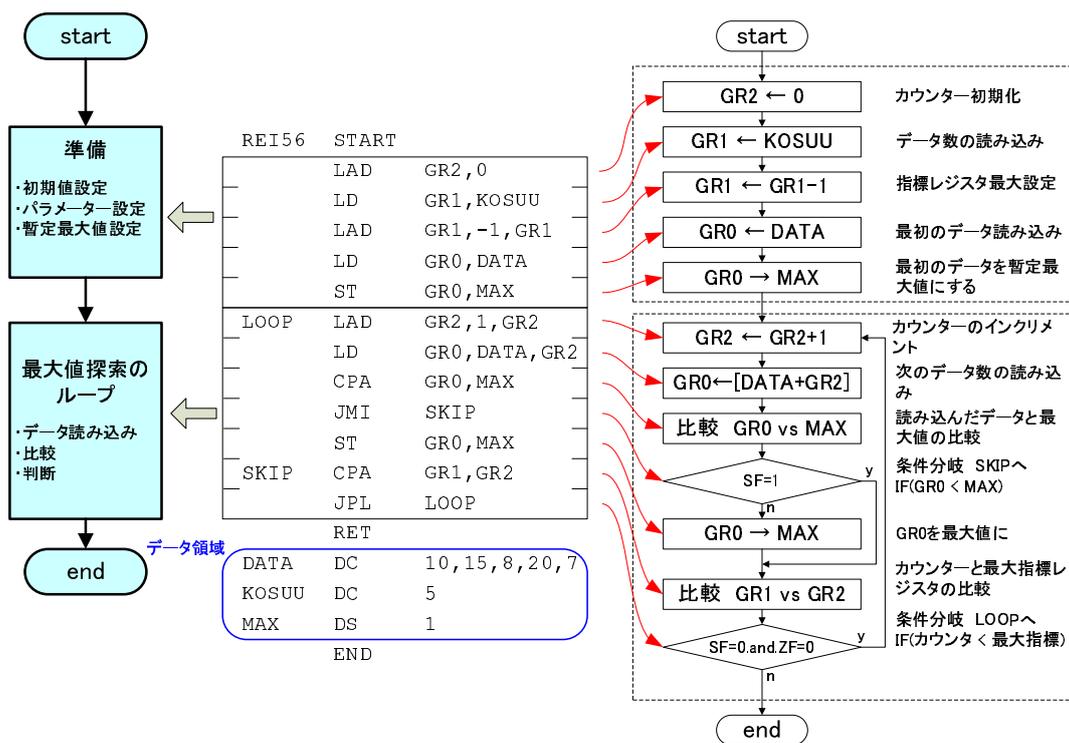


図 2: 教科書の List5-6 のプログラムの構造とフローチャート

3 [例題 7] 繰り返し処理とサブルーチン

3.1 教科書の例

教科書の [例題 6] のプログラムの動作内容は、[例題 5] と全く同じである。ただし、最大値を探索する部分をサブルーチンにして、プログラムの内容を分かりやすくしている。

3.2 サブルーチン

プログラムは、分かりやすく書かなくてはならない。分かりにくいプログラムはメンテナンスが大変である。ここでは、最大値を探す機能をサブルーチンとして分割している。

実際、サブルーチンを作成するにもっとも気にかけることは、データの受け渡しである¹。メインルーチンからサブルーチンに、ある処理を依頼するのであるが、そのためにはデータが必要である。メイン → サブ、メイン ← サブと 2 通りある。高級言語ではいろいろな方法があるが、CASL II では汎用レジスターを使うのが一般的である。

例題のプログラムを例にすると、

- メインルーチンがサブルーチンに依頼している仕事の内容は、データの最大値を探すことである。
- そのために、メインルーチンはサブルーチンに、GR1 を用いて、データの個数を渡している。

3.3 プログラムの構造とフローチャート

データを取り扱うレジスターやラベルの内容を表 1 と同じである。また、プログラムのフローチャートを図 3 に示す。

4 練習問題

4.1 問題

4.1.1 繰り返しとサブルーチン

問題 (1) 1 ~ 1000 までの加算するプログラムの作成

- 1 ~ 1000 までの加算は、サブルーチンで実行すること。そして、繰り返し構造を用いた加算であること。
- 加算結果は、メモリーの適当な場所に核のすること。

¹受け渡しのデータのことを引数と言う。呼び出し側が渡すデータを実引数、呼び出された側が受け取るデータを仮引数と言う

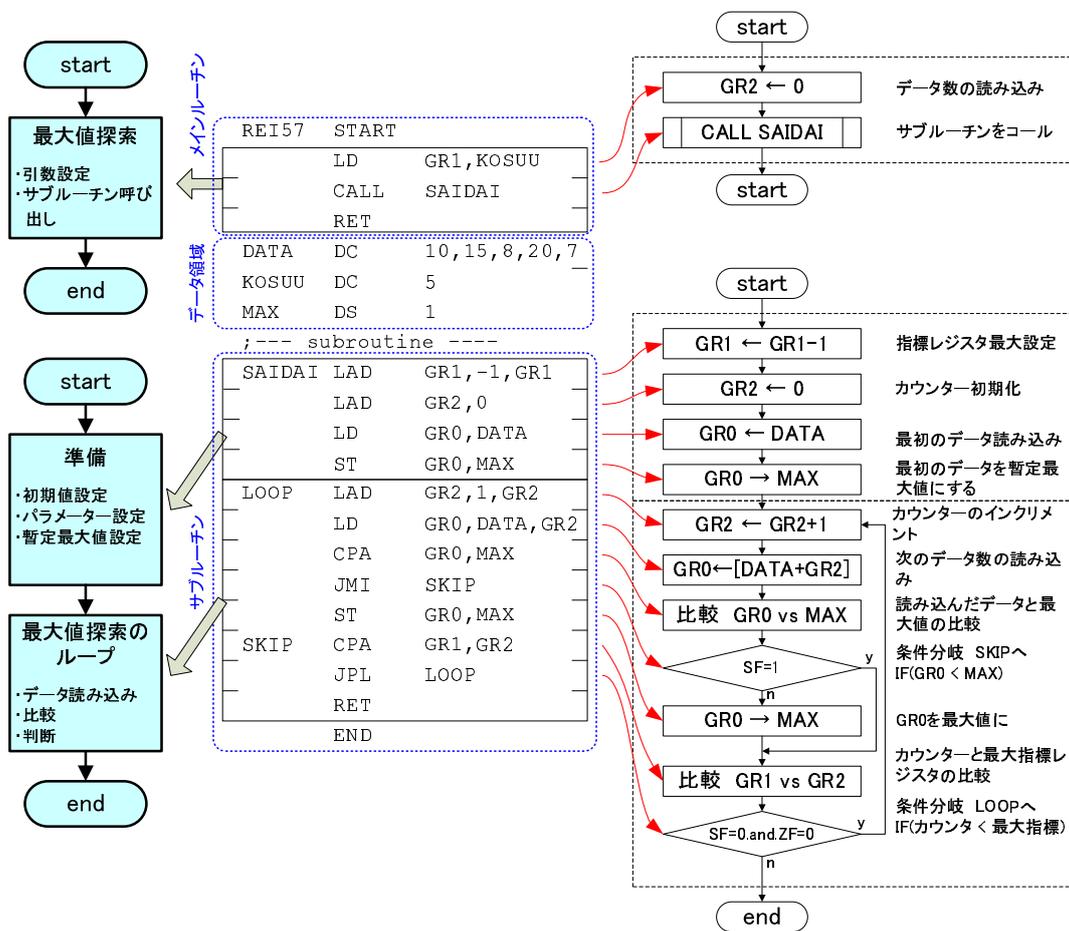


図 3: 教科書の List5-7 のプログラムの構造とフローチャート