

# CASL IIのプログラム例(その4)

山本昌志\*

2006年2月24日

## 1 復習と本日の内容

### 1.1 復習

前は、数値データを文字データに変換するプログラムを学習した。数値を文字コードに変換するためには、以下のようにした。最大の桁から文字コードに変換する。

1. 変換したい桁の値を求める。それには、数値をその桁で割る。(例  $23456 \div 10000 = 2$  余り 3456)
2. 商に#0030 を加算。(例 OR GR2,MOJI ;GR3: MOJI:#0030)
3. この加算した値が、文字コードになる。
4. 次の桁は、余り(例 3456)について同様の処理を行う。

要するに整数の文字を表す文字コードは、表1のようになっており、その差が#0030である。

表 1: 数字を表す文字のコード

文字	文字コード (16進数)
0	#0030
1	#0031
2	#0032
3	#0033
4	#0034
5	#0035
6	#0036
7	#0037
8	#0038
9	#0039

\*独立行政法人 秋田工業高等専門学校 電気工学科

## 1.2 本日の内容

本日は、前回の逆で、文字コードを数値に変換する。これは、マクロ命令 IN を使ってキーボードから数値を読み込んで、それを処理するプログラムに応用できる。

IN 命令で数値を読み込むが、それは文字コードでメモリーに格納される。その文字コードで格納された整数を実際の整数に変換する方法を学習する。

## 2 [例題 11] 文字データを数値データに変換

教科書の List5-11 のプログラムを例にして、文字データを数値データに変換する方法を学習する。

### 2.1 教科書の例

教科書のプログラムは、

- キーボードから数値を読み込む。読み込まれた数値は、文字コードになり、それをラベル BUFF が示すアドレスから格納される。読み込まれた文字数は、ラベル LEN に格納される。キーボードから入力する文字は、0~9 と '?' のみである。
- この、BUFF から格納された LEN 個の整数を表す文字列を整数に変換する。
- 変換された整数を汎用レジスター GRO 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.118-119 の List5-11 に示されている。

### 2.2 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 1 に示す。このプログラムの内容については、教科書に沿って説明する。

まずは、教科書 p.115 の文字を格納するメモリー領域を

```
BUF DS 256
```

が気になるところである。この 256 とはいったい何かである。256=2<sup>8</sup> である。これはコンピューターでよく使われる整数で、多分この場合は 1 レコードの値と思われる。1 回に読み込まれる最大文字数のつもりであろう (推測)。通常、このようにするとデータ転送の効率が良いのである。ただ、COMMET II にはそんな仕様は無いので気にする必要はない。実際に使われている機器のアセンブラのプログラムを書く場合は、注意が必要である。

後は教科書に沿って説明する。

表 2: メインルーチンの汎用レジスターとメモリの内容

GR0	最終的に、読み込んだ整数になる
GR1	桁の重みを読み出すときに使うインデックスレジスター
GR2	処理の桁を示し、先頭の桁 (符号の時もある) がゼロで、1 の位が LEN-1 となる
GR3	処理している桁の整数値
GR4	その桁の値 (桁を表す整数 × その桁の重み)
DATA	調べるデータの先頭アドレス。
BUFF	読み込まれた整数を文字コードとして格納している先頭のアドレス。
LEN	読み込んだ文字数。

表 3: サブルーチン MULTI の汎用レジスターの内容。このサブルーチンでは  $GR4 \times GR3 \rightarrow GR4$  を計算している。

レジスター	実行前	実行後
GR3	乗算する値 (カウンターとして使う)	0
GR4	乗算する値	除算の結果

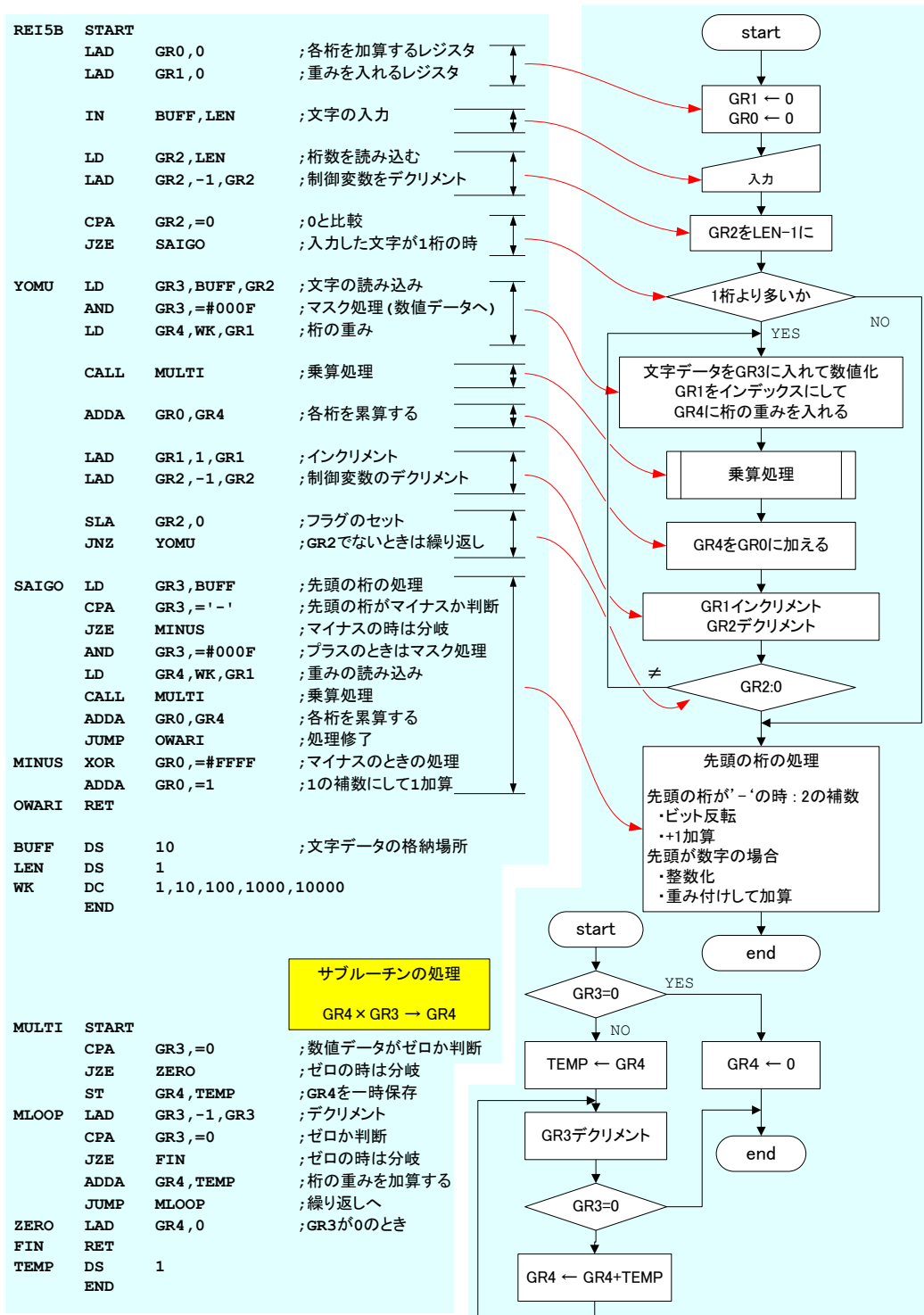


図 1: 教科書の List5-11 のプログラムの構造とフローチャート