

CASL IIのプログラム例(その3)

山本昌志*

2004年12月24日

1 [例題4] 論理演算とアドレス修飾

教科書の List5-4 のプログラムを例にして、論理演算とアドレス修飾について説明する。

1.1 論理演算

1.1.1 教科書の例

教科書のプログラムは、

- ラベル A,B に #0030, #009F が格納されている。
- ラベル ANS から、3 語このプログラムで確保されている。
- ANS から確保された 3 語の領域に、A AND B と A OR B、A XOR B の演算結果を格納せよ。

と言う問題を解く、プログラムである。

このようなプログラムを作成するために必要なことは、

- データ領域
 - － 演算の対象データ (#0030, #009F) をラベル (A, B) を指定してメモリーに書き込む。
 - － 演算結果を書き込む領域をラベル (ANS) を指定して、確保する。
- 命令領域
 - － 演算対象データをレジスターにコピー
 - － 演算の実行
 - － 計算結果の格納

である。

*国立秋田工業高等専門学校 電気工学科

1.1.2 アドレス修飾

大まかなプログラムの流れは、分かった。また、論理演算も解くに説明することもないだろう。演算対象のデータのそれぞれのビット毎の論理和 (OR) と論理積 (AND)、排他的論理和 (XOR) を計算しているだけである。

プログラムの命令領域とデータ領域は、図 1 のようになるだろう。プログラムの書き方によっては、こうならないこともあるが、通常はこのようになる。

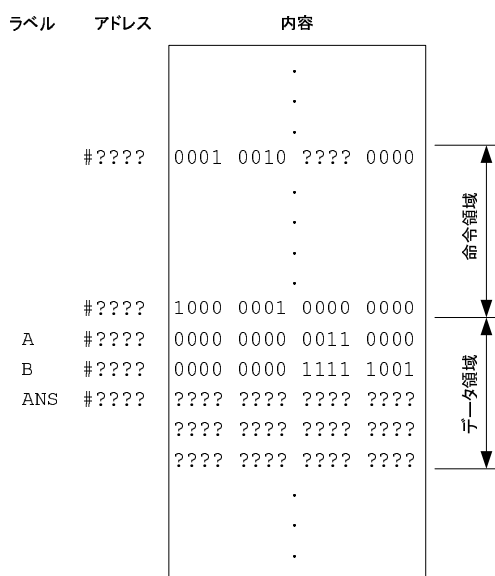


図 1: 教科書の List5-4 のプログラムを実行する場合のメモリ構造。図中の?は値はあるが、不明を示している。

この場合、プログラムのデータ領域にアクセスする事を考える。ラベル A や B は簡単で、ラベル名を示せば良い。ラベル名はアドレスを示すからである。問題は、結果を格納する領域である。このアドレスは、3 つ続いて確保されているが、先頭だけ ANS とラベル名がある。残りの 2 つの表し方である。これらのアドレスは、ANS+1 と ANS+2 である。ANS のアドレスにオフセットの値を加算するのである。

プログラムで使うメモリのアドレスは、ANS+オフセットで、オフセットは、0,1,2 とすれば良い。論理和の結果を ANS+0、論理積の結果を ANS+1、論理和の結果を ANS+2 に格納する。プログラムでは、オフセットの 0,1,2 を GR2 に入れておき、

```
ST GR1,ANS,GR2
```

と書く。演算の結果 (GR1) の値が、ANS にオフセット値 (GR2) を加えたアドレスに格納される。

ここで、使っている GR2 のように、1 つずつ値が増加するものをカウンターと呼ぶことがある。これを使うためには、

- カウンターの初期化。ここでは、GR2 をゼロに設定する。

```
- CASL では、LAD GR2,0
```

- カウンターのインクリメント。カウンターの値を 1 増加させる。

– CASL では、LAD GR2,1,GR2

とする。このテクニックは、重要である。内容をよく理解する必要がある。

1.2 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 2 に示す。

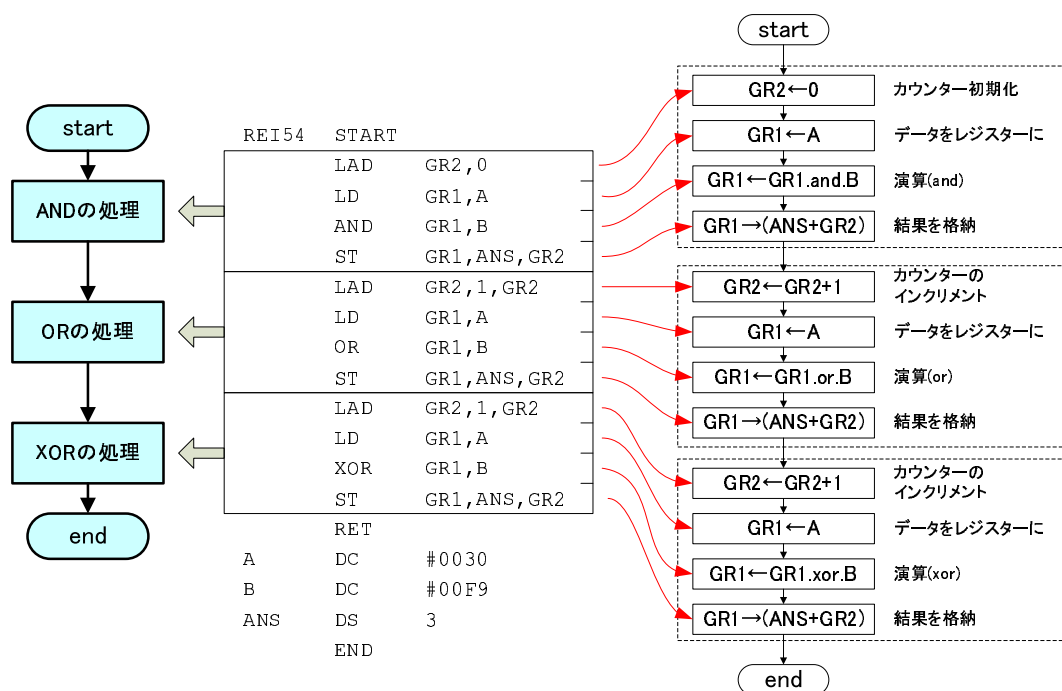


図 2: 教科書の List5-4 のプログラムの構造とフローチャート

2 [例題 5] シフト 演算

2.1 積の演算

積 (かけざん) の演算を行うとき、シフト命令を使えば効率の良いプログラムができる。ビットシフトを用いると積の演算ができる理由は以前述べているが、忘れた人もいるので、もう一度、説明する。

シフト命令を使った積の演算は、小学生のときに学習をした筆算の掛け算と同じである。たとえば、 34×24 を計算する場合、筆算は $34 \times (2 \times 10^1 + 4 \times 10^0)$ と分解したはずである。そうして、次の手順でこの除算を行ったはずである。

1. 34×2 を計算し、1 桁ずらす (10 倍する)。

2. 34×4 を計算する。

3. 先の計算結果を合計する。この合計 816 が 34×24 の計算結果である。

同じことを 2 進数で行う。これがコンピューターによる乗算である。先ほどと同じ計算 (32×24) を行う。これを 2 進数で表現すると、

$$(100010)_2 \times (11000)_2 = (100010)_2 \times (1 \times 2^4 + 1 \times 2^3)$$

となる。これを先ほど同様の手順で計算する。

1. 掛け算は 1 倍なので計算する必要が無く、最初に $(100010)_2$ を 4 桁左にずらす (ビットシフト)。すると、 $(1000100000)_2$ となる。
2. 次に $(100010)_2$ を 3 桁左にずらす。すると、 $(100010000)_2$ となる。
3. 先の計算結果を合計すると、 $(1100110000)_2$ となる。これは、10 進数の 816 である。

シフトと加算命令でかけ算ができることが分かったはずである。

今回の問題の用に分数の場合でも、

$$\begin{aligned} 0.75 &= \frac{1}{2} + \frac{1}{4} \\ &= (2^{-1}) + (2^{-2}) \end{aligned} \tag{1}$$

と分解する。右に 1 ビットシフトさせたものと、右に 2 ビットシフトさせたものを加算すれば良い。教科書のように

$$0.75 = 1 - (2^{-2}) \tag{2}$$

と分解するのは一般的ではない。

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 3 に示す。。

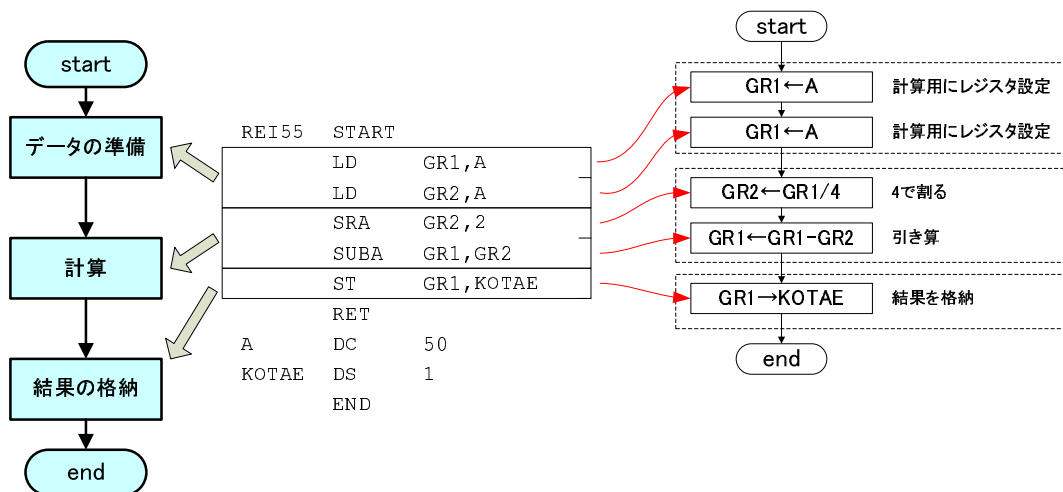


図 3: 教科書の List5-5 のプログラムの構造とフローチャート

3 課題

課題を課すので、レポートとして提出すること。課題内容は、以下の通り。

3.1 問題

3.1.1 アドレス修飾とカウンター

問題 (1) データの格納 (I)

- データ領域を 3 ワード確保する。
- 確保された領域に、アドレス修飾を利用して、1, 2, 3 と整数を格納する。

問題 (2) データの格納 (II)

- データ領域を 3 ワード確保する。
- 確保された領域に、アドレス修飾を利用して、2, 4, 6 と整数を格納する。

問題 (3) データの格納 (III)

- データ領域を 100 ワード確保する。
- 確保された領域に、アドレス修飾を利用して、2, 4, 6, ..., 200 と整数を格納する。ヒント：ジャンプ命令を上手に使うこと。

3.1.2 シフト演算

問題 (1) データを 8 倍

- ラベル名 DATA が示すメモリーの領域に $(00FF)_{16}$ の値を格納する。
- シフト命令を利用して、この値を 8 倍する。
- 8 倍された値は、ラベル名 KEKKA が示す領域に格納する。

問題 (2) データを 1/16 倍

- ラベル名 DATA が示すメモリーの領域に $(30000)_{10}$ の値を格納する。
- シフト命令を利用して、この値を 1/16 倍にする。
- 1/16 倍された値は、ラベル名 KEKKA が示す領域に格納する。

問題 (3) データを 5.75 倍

- ラベル名 DATA が示すメモリーの領域に $(100)_{10}$ の値を格納する。
- シフト命令を利用して、この値を 5.75 倍にする。
- 5.75 倍された値は、ラベル名 KEKKA が示す領域に格納する。

3.2 提出要領

期限 1月14日(金)PM5:00 まで

用紙 A4

提出場所 山本研究室の入口のポスト

表紙 表紙を 1 枚つけて、以下の項目を分かりやすく記述すること。

授業科目名「電子計算機」

課題名「CASL II プログラム演習 (その 3)」

3E 学籍番号 氏名

提出日