COMET IIのレジスター

山本昌志*

2004年5月24日

1　これまでの復習と本日の内容

1.1　これまでの復習

コンピューターを構成する最も重要な要素は、

- Central Processing Unit (CPU:中央処理装置)
- メインメモリー (main memory:主記憶装置)。単にメモリーと呼ぶことも多い。

です。これまでは、メインメモリーの中でのデータ (数値、文字) の格納方法を学習しました。次のようなことを理解しなくてはなりません。

- COMET IIでは、16ビットを1ワード (1語) と言い、この単位でデータの処理を行います。
- メモリーには、1ワード毎にアドレスがついています。
- 数値も16ビットで表現します。符号付整数の場合は、それで表すことができる範囲は、以下の通りです。
  - 正の数の絶対値の最大値は、$(0111\ 1111\ 1111\ 1111)_2 = (2^{15}\text{-}1)_{10}=(32767)_{10}$
  - 負の数の絶対値の最大値は、$(1000\ 0000\ 0000\ 0000)_2$ です。これは第15ビットが1なので負の数で、2の補数表示です。したがって、その絶対値を求めるためには、ビット反転を行い、1を加算すればよい。したがってこれは、$\text{-}(2^{15})_{10}=\text{-}(32768)_{10}$ を表します。
- 符号無整数の場合は、以下の通りです。
  - 表現可能な最小値は、$(0000\ 0000\ 0000\ 0000)_2 = (0)_{10}$ です。
  - 表現可能な最大値は、$(1111\ 1111\ 1111\ 1111)_2 = (2^{16}\text{-}1)_{10}=(65535)_{10}$ です。
- 数値と異なり、文字にはそれぞれ、番号をつけて区別します。文字とそれに対応する番号は、規格 JIS X0201 ラテン文字・片仮名用8単位符号で決まっています。

*国立秋田工業高等専門学校　電気工学科

- この番号は、8 ビットなので、最大 256 文字しか使えません。数字とアルファベットと片仮名と記号を表すのであれば十分です。漢字は、使えません。
- COMET II の 1 ワード 16 ビットに対して、文字は 8 ビットしか使いません。COMET II では 1 ワードで 1 文字を表すため、16 ビットのうち上位 8 ビットは 0 として、下位 8 ビットで 1 文字分を表します。例えば、アルファベットの Yama を表す場合、Y は $(59)_{16}$、a は $(61)_{16}$、m は $(6D)_{16}$、という番号がついているので、COMET のメモリーには、次のように格納されます。ただし、アドレスの実際の割り当ては、OS が決めます。

図 1: 文字列”Yama”のメモリーへの格納

- 数値と文字では、メモリーの中身は異なります。例えば、数値の $(9)_{10}$ と文字の”9”は、以下のようになります。文字の”9”は、JIS X0201 では、$(39)_{16}$ です。

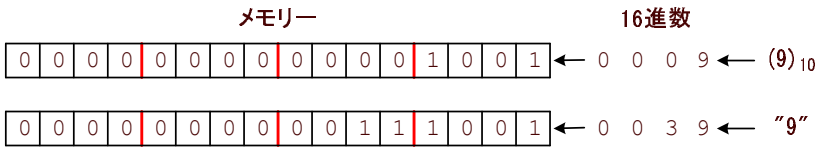

図 2: 数値の $(9)_{10}$ と文字”9”のメモリーへの格納

- メモリーの中身を見ると、それが数値なのか文字なのか、判断できません。命令毎に数値を扱うのか、文字を扱うのか決まっています。以降の、学習で分かるでしょう。

1.2 本日の内容

本日は、レジスタについて学習します。これは、CPU 内のデータの記憶する場所と考えてください。教科書の P.15〜P.20 の指標レジスタの前までです。ここまでを中間テストの範囲とします。

2　レジスタとは何か

2.1　レジスタとは何か

レジスタを一言で言うと、CPU 内の記憶装置みたいなものです。メモリーと同じで、いろいろなデータを記憶させます。メモリとレジスタの違いは、どこにあるのでしょうか?。以下のような違いを列挙できます。

- レジスタは CPU 内部にあり、データを加工するために一時的に、記憶させる。また、加工結果も記憶する。
- メモリに比べて、レジスタの記憶領域は小さい。COMET II の場合、メモリーのアドレス数は 65536 個あるに対して、レジスタは 20〜30 個程度です。
- メモリは番地を指定して目的のデータにアクセスする。一方、レジスタは名前を指定して、目的のデータにアクセスする。
- 現実の装置の場合、CPU のデータのアクセススピードは、レジスタの方がはるかに早い (C 言語ではレジスタを使ったプログラムができる)。

2.2　コンピューターはどのようにプログラムを実行するか

コンピューターのプログラムは、データと命令から構成されます。この命令とデータは、実行時に主記憶装置 (メインメモリ) に格納されます。この格納の動作をロードと言います。これらのプログラムは、CPU 内部のレジスタに読み込まれ、処理されます。実際のコンピューターでのプログラムの動作順序は、次の通りです。

1. 補助記憶装置 (ハードディスク等) からプログラムがメインメモリーにロードされます。この指令は、Operating System(OS) が出します。
2. メインメモリーに格納されたプログラムの指示に従い、CPU が動作します。その動作は、
   (a) CPU がメインメモリーから命令を取り出します。命令を取り出すアドレスは、CPU のプログラムレジスタに書かれています。
   (b) 取り出した命令は、CPU 内の命令デコーダーにより、命令の内容を解析されます。
   (c) 解析された命令は、論理演算装置 (ALU:arithematic logic unit) により、演算が実行されます。
   (d) 演算結果は、各種のレジスターに格納されます。
   (e) プログラムレジスタの値を再設定します。
   (f) 以上の動作をプログラム終了まで繰り返します。

   です。

このプログラムの実行方法からも、CPU の中にもデータを記憶する場所が必要と理解できるでしょう。その CPU 内の記憶装置をレジスタといいます。わざわざ CPU 内に作らなくても、メインメモリーの一部を

使えば良いのでは、と考える人も居るかもしれません。たぶんそれでもコンピューターはできるでしょうが、今よりも複雑になると思います。また、CPU とメモリーのデータの交換が増えて、動作が遅くなるでしょう。

CPU と主記憶装置は、図 3 のような関係です。CPU は主記憶装置のアドレスを指定することにより、主記憶装置に格納されているデータを引き出します。そして、それはレジスタに記憶され、その中身に従い、処理されます。処理された結果ももちろん、レジスタに記憶されます。レジスタの中身を主記憶装置に戻すことにより、データの加工が完了します。

レジスタもデータなどを蓄えるので、メインメモリー同様、記憶装置の一種です。しかし、それぞれ、役割が異なります。

- 主記憶装置
  - CPU とは独立です。
  - プログラムを格納します。
  - データも格納します。
- レジスタ
  - CPU の構成部品のひとつです。
  - 演算の対象や演算結果を格納します。
  - 主記憶装置のアドレスを格納するレジスタもあります。

要するに主記憶装置は、いろいろなデータ (命令もデータの一種と考える) を蓄えるファイルキャビネットのようなものです。一方、レジスタは、実際に CPU がデータを加工するときに一時的に記憶する場所と考えてください。

C 言語や FORTRAN のプログラムでは、主記憶装置のデータを加工して、書き換えているように思いますが、実際は、それらを加工する場合、レジスタが一時的にデータを記憶し、それを CPU が加工して、主記憶装置に戻しています。

3 COMET II のレジスタ

図 3 のうち、プログラマが注意を払うべきものは、

- 主記憶装置
- レジスタ

です。今後アセンブラでプログラムを書いてみると分かりますが、制御装置や演算装置について、あまり注意を払う必要はありません。COMET II のレジスタを表 1 にまとめておきます。以降、それぞれのレジスタについて、説明します。

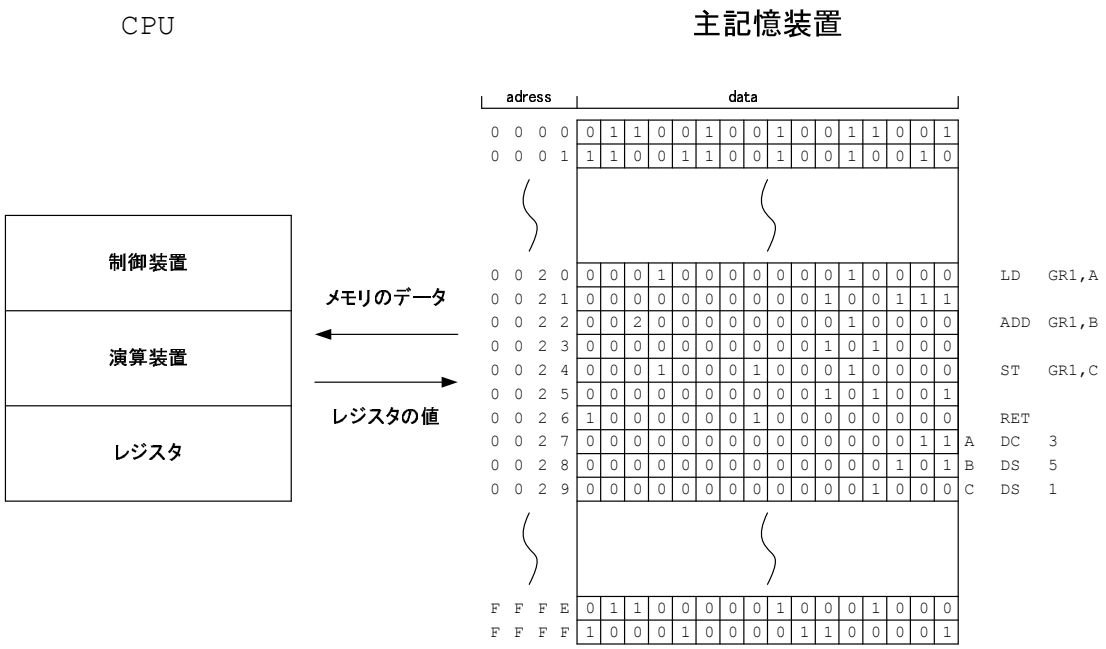

図 3: CPU と主記憶装置の関係

表 1: CASL II のレジスタ

記号	語源	日本語	機能
GR	General Register	汎用レジスタ	計算等に用いる。また GR1〜GR7 は指標レジスタとしても使われる。
SP	Stack Pointer	スタックポインタ	スタック領域の最上段のアドレスを保持する。
PR	Program Register	プログラムレジスタ	次に実行する命令のアドレスを保持する
FR	Flag Register	フラグレジスタ	演算結果の状態を保持する

3.1 汎用レジスタ

これは、算術や論理、比較、シフト演算を実行するときに使います。GR0〜GR7 までの 8 個用意されています。あとは、教科書の通りです。

- 汎用レジスタは、8 個用意されています。
- 汎用レジスタは、16 ビットです。メインメモリーのデータのビット数とおなじです。

3.2 プログラムレジスタ

プログラムカウンターと呼ばれることもあります。このレジスタの値は、プルグラムが次に実行する命令語の先頭番地です。したがって、

- 必要なプログラムレジスタは、1 個です。
- プログラムレジスタは、16 ビットです。アドレスのビット数と同じ。

となります。

プログラムを事前に主記憶装置に格納して、プログラムレジスタ PR の値によって、プログラムを構成する命令を 1 つずつ取り出して、処理を行います。このような方式を逐次制御方式と言ったり、プログラム内蔵方式 (stored program) と言ったりします。

3.3 フラグレジスタ

Flag Regisuter のフラグとは、旗のことです。サッカーの試合で、プレーの状態により旗を上げます。あれと同じです。コンピューターでは演算の結果により旗を上げます。

COMET II には、1 ビットのレジスタが 3 個あります。演算結果によって、それらのレジスタの値がセットされます。セットされる内容は、教科書 P.18 の表 2.4 の通りです。主に、このレジスタは、実行順序を変更、分岐命令に使われます。

- フラグレジスタは、3 個あります。それで、計算結果の状態を表します。
- 各レジスタは、旗の上げ下げなので、1 ビットです。

あとは教科書の説明通り。

3.4 スタックポインタ

メインメモリーの一部を CPU が専用の記憶領域として使います。そのときのメインメモリーのアドレスを示します。したがって、

- 必要なプログラムレジスタは、1 個です。
- プログラムレジスタは、16 ビットです。アドレスのビット数と同じ。

となります。

これは、ここでは少し早すぎますので、実際に使うときに説明します。

3.5 指標レジスタ (index register)

これは、特殊なレジスタで、ハードウェアーは汎用レジスタが兼ねます。汎用レジスタのうち GR1〜GR7 をつかいます。GR0 を使わない理由、これはマシン語との関係で、後の授業で述べます。

- 指標レジスタは、汎用レジスタの 7 個が使えます。
- 指標レジスタは、16 ビットです。メインメモリーのデータのビット数と同じです。

教科書の図 2.5 の表現は分かりにくいので、具体例でその動作を示します。例えばクラス 40 人分の数学と英語と電子計算機のテストの点が、メモリに格納されており、それぞれの平均点を求めたい場合、指標レジスタを使うと便利です。このプログラムでは、それぞれの教科のクラスの合計点を計算するところが、重要です。指標レジスタを使う場合と使わない場合のフローチャートを図 4 に示します。

指標レジスタを使わないと、プログラムが大変でしょう。このように、指標レジスタを使うことにより、基準点からのアドレスを加算してそのデータにアクセスできます。このように、アドレスを操作することをアドレス修飾と言います。

実は、皆さんは、これと同じプログラムテクニックを FORTRAN の授業で学んだはずです。FORTRAN の配列と同じです。FORTRAN では分かりにくいのですが、C 言語の配列はまさにこれと同じことを行っています (実感できます)。

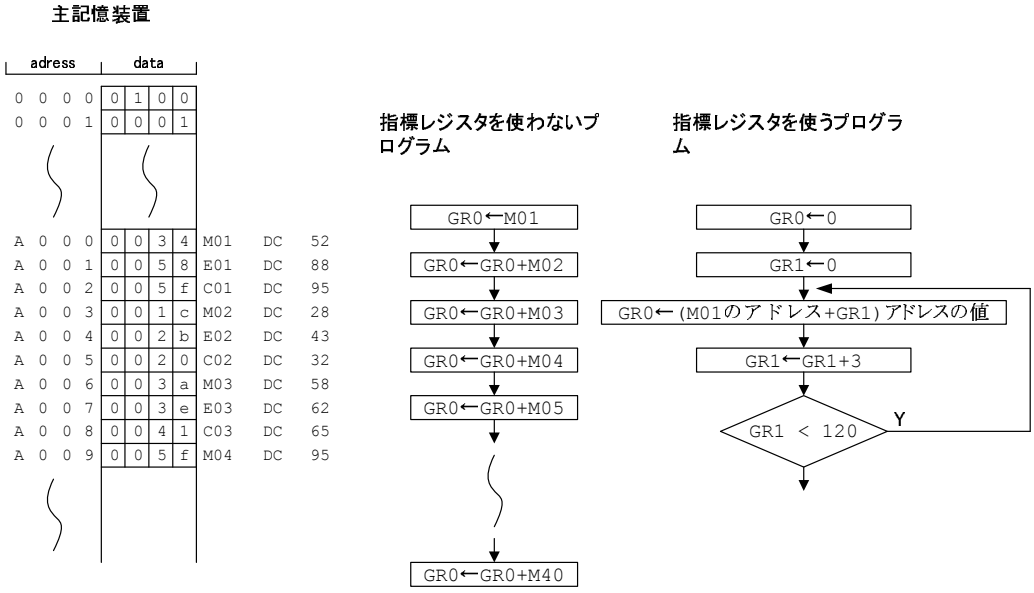

図 4: 指標レジスタを使った場合と使わない場合のプログラム。クラスの数学のテストの合計点を計算している。GR1 を指標レジスタとして使っている。