

### 本節の授業のテーマ

本日の事業のテーマは、以下のとおりです。

(1) レジスタとメモリ

- レジスタと主記憶装置(メインメモリ)
- COMET II の主記憶装置(メインメモリ)の構造
- COMET II のレジスタ

本日の授業のゴールは、以下のとおり。

- レジスタとメモリの違いが分かる
- 主記憶装置の構造が理解できる
- COMET II のレジスタの役割が理解できる。

## 0. 前回の復習

- COMET II のシミュレーターを紹介しました。
  - Infocas1
  - WCASL-II
- 主記憶装置(メインメモリ)は、データとアドレスがあります。アドレスを指定することによって、目的のデータにアクセスします。
- 主記憶装置のアドレスは 16 ビットです。したがって、10 進数では、 $(0)_{10}$  番地から  $(65535)_{10}$  番地までです。16 進数では、 $(0000)_{16}$  番地から  $(FFFF)_{16}$  番地までです。
- CPU 内部でデータを記憶する場所が、レジスタです。加工するデータを一時的に記憶したり、加工結果が格納されたりします。
- シミュレーターWCASL-II を使って、COMET II の動作を調べました。構造や以下の動作を確認しました。
  - メインメモリのアドレスとデータの関係。
    - アドレスは、16 ビット
    - データも 16 ビット
  - プログラムもデータもメモリの中に格納されている(ノイマン型コンピューター)。
    - メインメモリのデータをレジスタに渡す方法。
    - レジスタもアドレスも 16 ビット
  - プログラムレジスタ(PR)の動作の確認。
    - アドレスが 16 ビットなので、プログラムレジスタも 16 ビット
    - 次に実行する命令の先頭番地を格納している。
  - 計算結果が負になると、フラグレジスタが 1 になります。
  - 負の数は、2 の補数で表現されています。

## 1. レジスタと主記憶装置

コンピュータのプログラムは、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。

レジスタもデータなどを蓄えるので、主記憶装置同様、メモリの一種です。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。

CPU と主記憶装置は、図 1 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。

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。

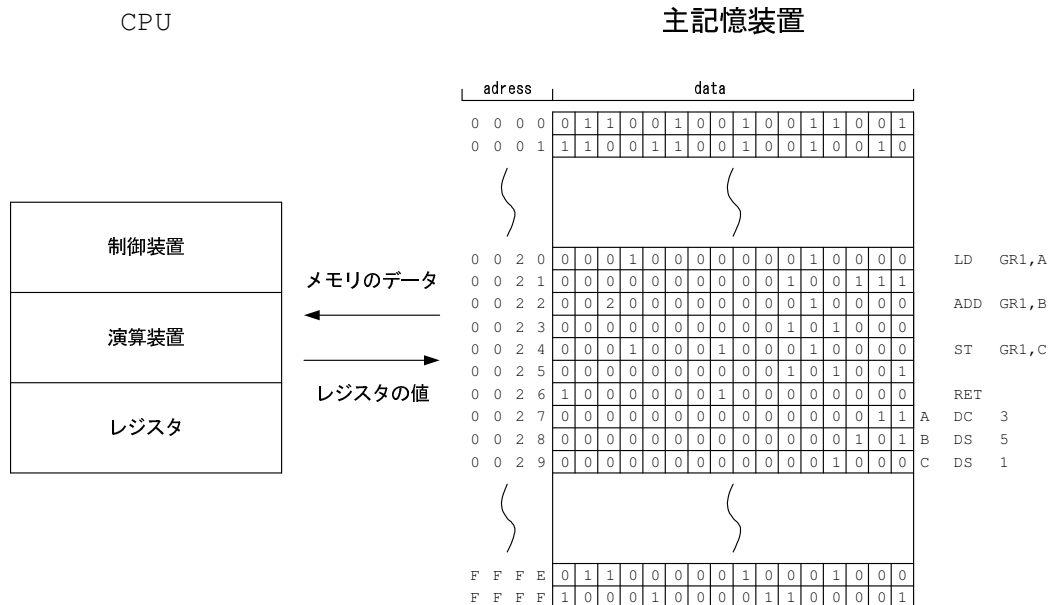


図 1 CPU と主記憶装置の関係

## 2. 主記憶装置(メインメモリ)の構造

主記憶装置にはアドレスとデータがあります。その概念は、図1に示した通りです。CPUは番地を指定することにより、目的のデータにアクセスできます。主記憶装置の物理的な構造は、

- アドレスは、16ビット。したがって、10進数では、 $(0)_{10}$ 番地から $(65535)_{10}$ 番地までです。16進数では、 $(0000)_{16}$ 番地から $(FFFF)_{16}$ 番地までです。
- データ16ビット毎に1個のアドレスが割り当てられています。

となっています。

主記憶装置に格納されるデータは16ビットが基本単位になっており、それを1ワードといいます。CASL IIの命令も1ワード単位で構成されます。命令によって、その命令語長は、1または2ワードとなっています。

処理するデータも、1ワード単位です。したがって、データを記憶するレジスタは16ビットになっています。データの処理の単位は、16ビットということになります。これ以外の単位でデータを処理したい場合は、特別にプログラムを書く必要があります。

アドレスも16ビットですから、アドレスを記憶するレジスタも16ビットになります。

## 3. レジスタ

図1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。それぞれのレジスタについて、以降、説明します。

表1 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
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
PR	Program Register	プログラムレジスタ	次に実行される命令のアドレスを保持
FR	Flag Register	フラグレジスタ	演算結果の状態を保持

### 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。

### 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、プログラムが次に実行する命令語の先頭番地です。

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

これは、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。

あとは教科書の説明どおり。

### 3.4 指標レジスタ(index register)

これには、汎用レジスタGRのうちGR1~GR7をつかいます。教科書の図2.5の表現は分かりにくいので、具体例を示します。

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2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が、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)。

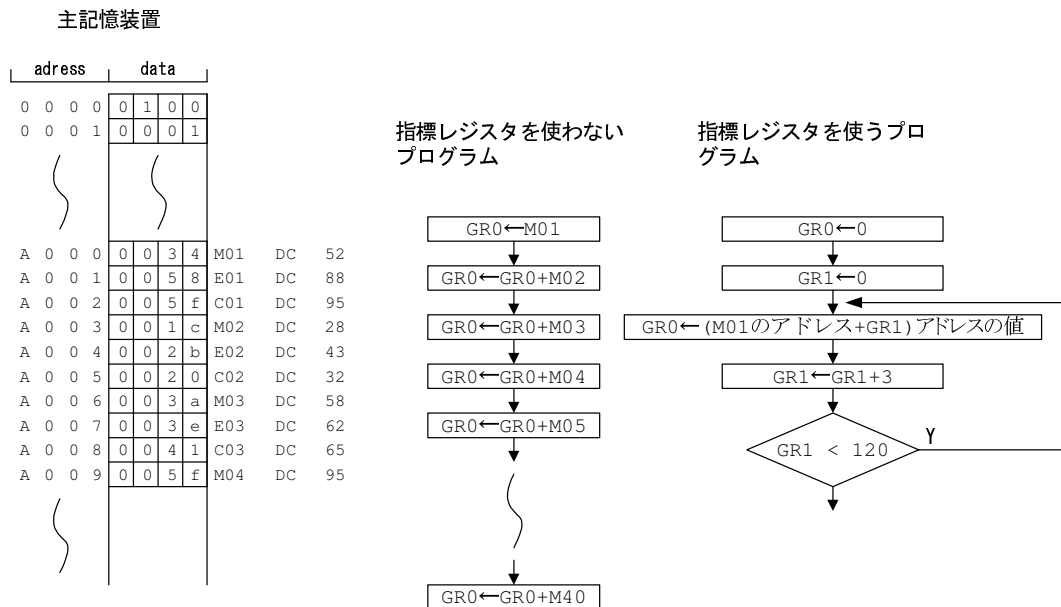


図2 指標レジスタを使わない場合と使った場合のプログラム。数学のテストの点のクラスの合計点を計算している。GR1を指標レジスタとして、使っている。

### 3.5 スタックポインタ

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。