電子計算機 **3E**

3E　2003.5.26

いよいよ、アセンブラ言語の勉強に入ります。今までは、アセンブラ言語を学ぶための、基礎的な学習です。では、いったいなぜアセンブラ言語を勉強するのでしょうか?。以下のような理由が考えられます。この中で、1 つでも当てはまるものがあれば、勉強の価値はあるでしょう。

- コンピューターの動作原理を知りたい。
- コンピューターの基礎の勉強。コンピューターの動作の勉強のため。
- 他の言語、例えば C 言語を勉強するときに参考になる。
- 実行速度の速いプログラムが作成できる。
- アセンブラを知っているとかっこいい。
- 将来ハードウェアーの基礎的な設計で必要。
- 将来、OS やコンパイラのプログラマーになるためには、勉強をしておく必要がある。
- なんか、コンピュータに直接命令を書いているようで、面白そうだ。
- 情報処理技術者試験の科目にあるから

本節の授業のテーマ

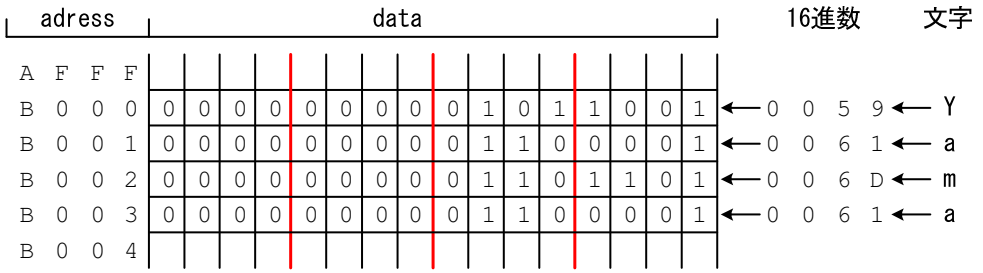
本日の事業のテーマは、以下のとおりです。

(1) COMET II の動作
- CPU とメモリーの関係
- メモリーのデータとアドレス
- レジスター

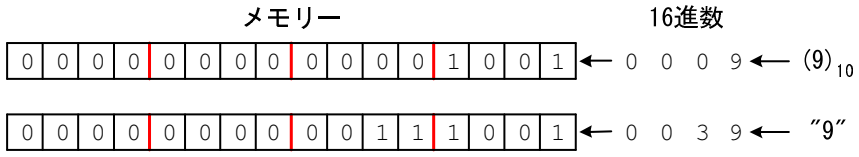
本日の授業のゴールは、以下のとおり。
- COMET II の動作が分かること。コンピューターの動作原理の理解。
- メモリーのアドレスとデータの意味がわかること。
- レジスターの動作が分かること。

0. 前回の復習

- 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します。
- 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 ビットの構成である。符号ビットがある場合の最大値は、以下の通りです。
  - 正の数の絶対値の最大値は、$(0111\ 1111\ 1111\ 1111)_2 = (2^{15}-1)_{10} = (32767)_{10}$
  - 負の数の絶対値の最大値は、$(1000\ 0000\ 0000\ 0000)_2 = (2^{15})_{10} = -(32768)_{10}$
- 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。
- 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 ビットしか使居ません。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。



- 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。



- 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. シミュレーター

1.1 シミュレーターの紹介

いよいよ、アセンブラ言語の勉強に入ります。今までは、アセンブラ言語を学ぶための、基礎的な学習です。

フリーウェアーのシミュレーターを紹介しておきます。Windos で動作しますので、ダウンロードして学習に役立ててください。また、中間試験の間に、情報教育ルームのパソコンにインストールすることも考えています。

- Infocasl
  教科書の p.187 以降に説明してあります。シンプルで使いやすいです。
  URL　　http://www.rs.kagu.tus.ac.jp/~infoserv/j-siken/infocasl/

- WCASL-II
  ハードウェアー、COMET II の動作のシミュレートでき、コンピューターの動作が良くわかります。本日の授業で使ったソフトウェアーです。
  URL　　http://www.ics.teikyo-u.ac.jp/wcasl2/

1.2 WCASL-II

COMET II の動作を理解するために、WCASL-II[1]を使います。この COMET II シミュレーターについて紹介します。

プログラムを書いて、COMET II シミュレーターを実行させると図 1 の画面になります。これが COMET II のブロック図です。通常のコンピューターの CPU とメモリ(主記憶装置)とほとんど同じです。ただし、現在使われている CPU の機能よりは、かなり単純化されています。その分、コンピューターの本質を学ぶには良いでしょう。

このシミュレーターで使われているレジスタを表 1 にまとめておきます。ただし、COMET II の仕様(教科書 P207～)に無いものもあります。というのは、アセンブラ言語の仕様にはなくても差し支えないが、実際のハードウェアーを構成する場合、通常必要なものがこの WCASL-II にはあります。COMET II の仕様に書かれていなくて、WCASL II にあるものは、表 1 の中で

IR　MAR　MDR

です。これらの 3 個のレジスタは、皆さんがアセンブラ CASL II を記述する場合は、関係ありません。また、仕様にも書かれていないので注意してください。

また、レジスタ以外のモジュールを表 2 に書き出します。これらのモジュールも、COMET II や CASL II の仕様には書かれていません。したがって、アセンブラ言語を記述する場合、これらを考える必要はありません。ただし、ハードウェアーにはこれらが必要なので、シミュレーターには、備え付けられています。

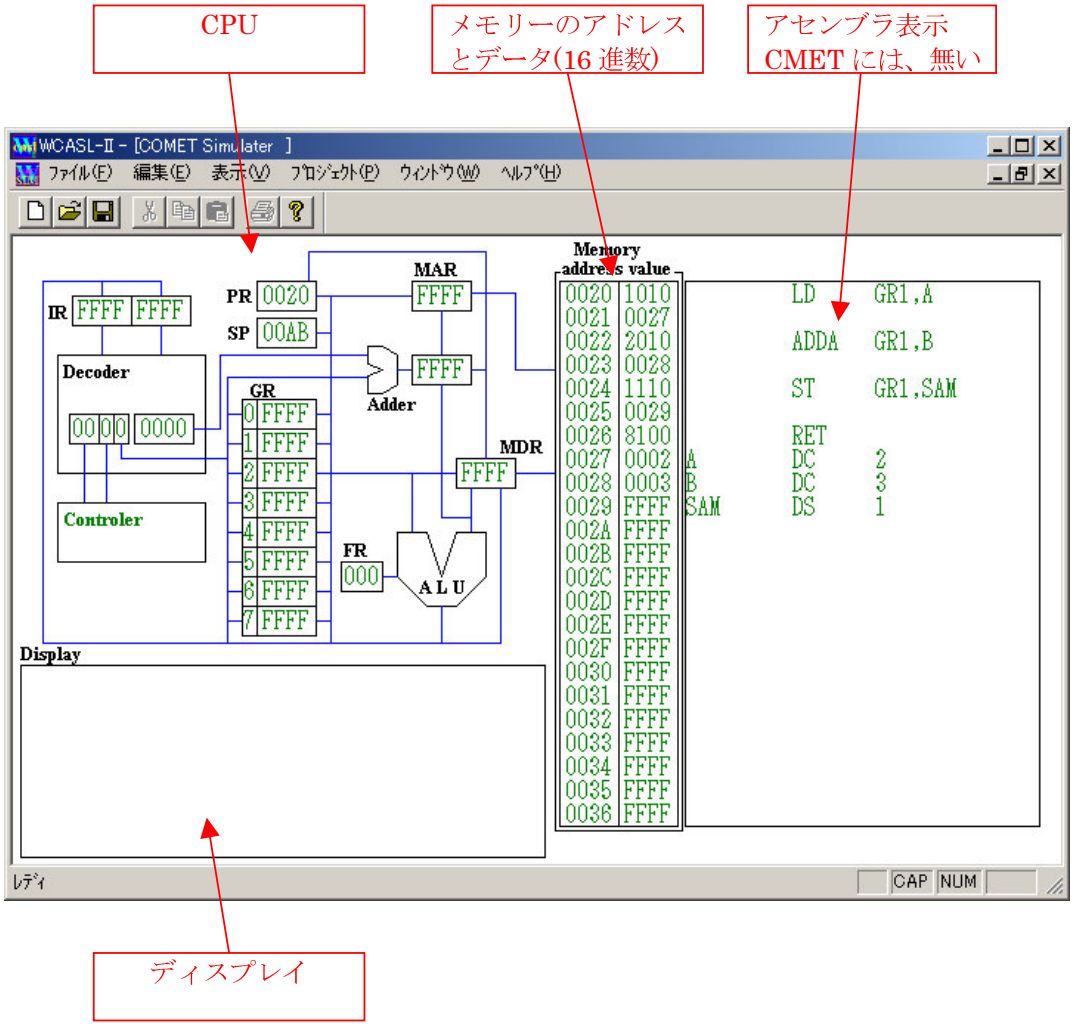


図 1 WCASL-II が示す COMET II のブロック図

表 1 WCASL-II のレジスター

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
IR	Instruction Register	命令レジスタ	命令そのものを格納
MAR	Memory Address Register	メモリアドレスレジスタ	メモリを読み書きする際に，アクセスするアドレスを格納
MDR	Memory Data Register	メモリデータレジスタ	メモリとのデータのやり取りをする際のデータを格納

表 2 WCASL-II のレジスター以外のモジュール

記号	語源	日本語	機能
ALU	Arithmetic and Logical Unit	演算装置	算術演算と論理演算他を実施
Decoder		命令解読器	命令を解読するモジュール
Controler		制御装置	命令の実行のための制御を行うモジュール
Bus		バス	レジスタや ALU などの間をデータが通る道
Adder		アドレス加算器	命令のアドレス部とインデックスレジスタの内容から有効アドレスを求める加算器
Memory		記憶装置	プログラムやデータを記憶しておく場所

2. メモリ(主記憶装置)の構造

2.1 アドレスとデータ

メモリ(主記憶とも言う)にはアドレスとデータがあります。その概念を図 2 に示します。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。一方、CPU 内の記憶装置であるレジスタは、名前によりデータを指定してアクセスします。

COMET II のアドレスとデータは、以下のようになっています。

- 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 $(65535)_{10}$ 番地までである。16 進数では、$(0000)_{16}$ 番地から $(FFFF)_{16}$ 番地までである。
- 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。

2.2 メモリとレジスタ

レジスタもデータを記憶するのでメモリと同様の動作をします。それでは、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、レジスタの記憶場所は小さい。
- 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。

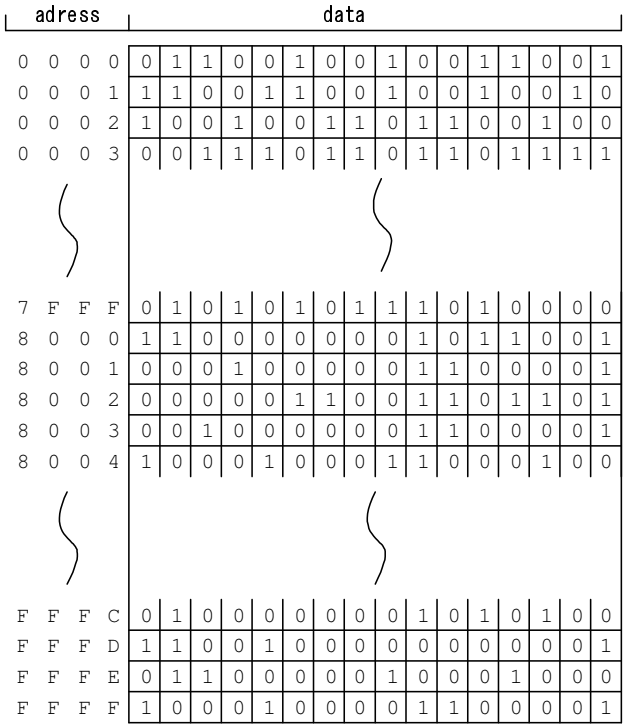


図 2 メモリ(主記憶)の概念。アドレスは 16 進数表示、データは 2 進数で表示している。

3. プログラムの実行

3.1 正の数の加算

教科書に載っている 3+5 を計算するプログラムを実演します。プログラムは、

```
PGM   START
      LD      GR1,A
      ADDA    GR1,B
      ST      GR1,C
      RET
A     DC      3
B     DC      5
C     DS      1
      END
```

です。このプログラムを実行させますので、以下に注意して、その動作を観察してください。

- メモリー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)。

・メモリーのデータをレジスタに渡す方法。
  ・レジスターもアドレスも 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

3.2 負の数の加算

つぎに、負の数の加算を実行させます。プログラムは、

```
PGM  START
     LD      GR1,A
     ADDA    GR1,B
     ST      GR1,C
     RET
A    DC      3
B    DC      -5
C    DS      1
     END
```

です。このプログラムを実行させますので、以下に注意して、その動作を観察してください。

・フラグレジスタが、1 になること。
・負の数は、2 の補数で表現されていること。

4. 参考文献

[1] WCASL-II については、以下を参考にした。
http://www.ics.teikyo-u.ac.jp/wcasl2/

ちょっと息抜き

教科書の p.17 の図 2.4 は、4 個所、間違いがあります。次回の授業までに、見つけてください。

ヒント　教科書 p213 の命令語の構成が理解できると、図 2.4 の主記憶の中の 2 進数データの意味がわかります。