

本日の授業のテーマ

- (1) なぜアセンブラ言語 CASLII を勉強するのか
 - ・ 基本情報処理試験
 - ・ コンピューターの仕組み
- (2) コンピューターの仕組み
 - ・ コンピューターの動作
 - ・ 主記憶装置のデータとは何か?

1 なぜ CASLII を勉強するのか?

年間の授業の半分が終了しました。期末のテストの成績が思ったよりも悪かったこともあるので、ここでもう一度学習の意義を考えましょう。なぜ、センブラ言語 CASLII を学習するか?。いろいろ考えられますが、少なくとも、以下の理由が考えられると思います。

- ① 基本情報処理試験の試験科目として勉強する。
- ② コンピューターの仕組みを理解する。
- ③ 将来、もっと高度なアセンブラ言語を学習するための下地。

本日は、そのうち最初の2つについて説明をします。

2 基本情報処理試験

情報技術者のスキルを試す試験として情報処理技術者試験¹というものがあります。それは、図1のような体系になっています。いろいろな試験がありますが、その基本となるのが、基本情報処理技術者試験です。将来のことを考えると、この試験に合格したほうが良い²と思います。中学生でも合格しているのやる気さえあれば合格できるはずです。

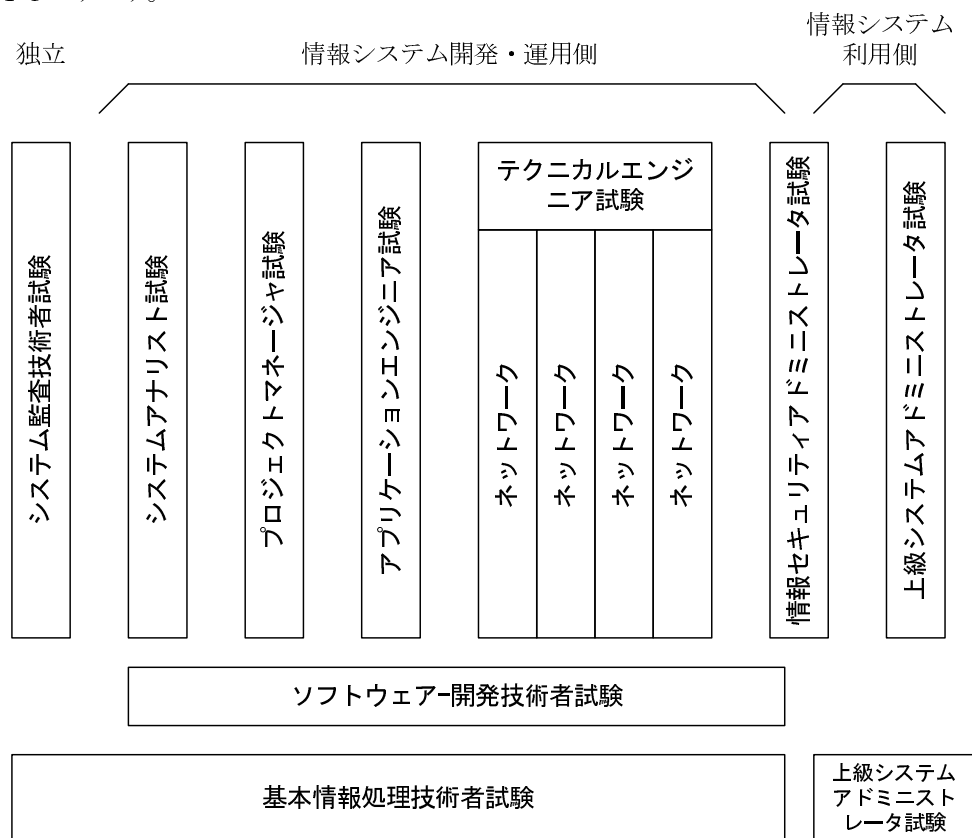


図1 情報処理技術者試験の体系

¹ 詳細は、<http://www.jitec.jp/index.html> を見よ。

² 足の裏についてご飯粒と同じで、食べないが、取っておいたほうが良いに決まっています。

この試験では、プログラムの知識を問う問題があり、その言語としてC言語、COBOL、アセンブラ言語、Javaが選択可能となっています。アセンブラ言語は、ここで学習しているCASL IIです。せっかくCASL IIを学習するのでですから、基本情報処理試験を受験してみたいはいかがでしょうか？。

基本情報処理試験の実施要綱は、以下の通りです。期待する技術水準を見ると、難しそうに思えますが、実際はそんなに難しくありません。中学生でも合格しているくらいですから。

期待する技術水準

情報技術全般に関する基礎的な知識を活用し、情報システム開発においてプログラムの設計・開発を行うとともに、将来高度な技術者を目指す者として、次の知識・能力が要求される。

- 1) 情報技術全般に関する基本的な用語・内容を理解している。
- 2) 上位技術者の指導のもとにプログラム設計書を作成できる。
- 3) プログラミングに必要な論理的思考能力を有する。
- 4) 一つ以上のプログラム言語の仕様を知っており、その言語を使ってプログラムを作成できる。
- 5) プログラムのテスト手法を知っており、テストを実施できる。

試験形式と試験時間

午前	午前
9:30～12:00 (150分)	13:00～15:30 (150分)
多肢選択式 80問必須	多肢選択式 13問出題／7問解答 プログラム言語はC、COBOL、CASL II、Javaから選択

試験の手続日程

試験実施時期	春期	秋期
試験実施日	4月第3日曜日	10月第3日曜日
案内書、願書の配布と受付	1月上旬から2月上旬まで	7月上旬から8月上旬まで
受験手数料	5,100円(税込み)	

秋田で受験できます。受験してみましょ。何回か受験すれば、そのうち合格するでしょう。

3 COMET II を通して学ぶコンピューターの仕組み

3.1 COMET II の構成

期末テストの成績が予想よりも随分悪かったことを反省しました。コンピューターの仕組みが分かっていないことがもっともテストの点を悪くした原因だと思います。コンピューターの仕組みが分からずに先に進むと、ますます分からなくなります。そこで、もう一度コンピューターの仕組みを説明します。

これは、アセンブラ言語を学ぶもっとも基本的なことです。どんな学問でも基本さえしっかり理解していれば、後はなんとかなります。ここで、もう一度、コンピューターの基礎を理解してください。

コンピューターの仕組みはいたって単純です。それを、図 1 のように COMET II の CPU と主記憶装置だけを考えることにより理解しましょう。どんな、コンピューターでもこの2つは必ず有ります。キーボードが無いとかディスプレイが無いとか、ハードディスクが無いコンピューターは存在します。しかし、CPU あるいは主記憶装置の無いコンピューターは存在しません。どんなコンピューターでもその基本的な仕組みは同じです。スーパーコンピューターであろうが COMET II であろうが同じ仕組みで動作しています。

まずは、それぞれの役割をもう一度、言います。

CPU データの加工(演算)をします。単純なことしか出来ませんが非常に高速に処理します。この演算の回路は、2年生の時に学習した論理回路(組み合わせ回路、順序回路)で出来ています。論理回路は任意の入出力の論理でもできますので、必要な処理が出来ます。

扱うデータは全て 16 ビットです。

単純な演算を高速に処理することは得意ですが、記憶は苦手です。記憶容量は小さく、図に示すように 10 個程度のレジスタに記憶します。それぞれのレジスタは役割が決まっております、

プログラムレジスタ (PR)

次に実行する命令のアドレスを示します。実行した命令が 2 語で構成されていれば、+2 加算されます。1 語であれば、+1 加算されます。³

汎用レジスタ (GR)

計算を実行するためのデータを格納します。

フラグレジスタ (FR)

計算結果の状態を示します。

スタックポインター (SP)

スタック領域の最上位のアドレスを示します。これは後で学習。

主記憶装置 命令や処理をするデータ(数字や文字)を記憶します。記憶する場所には、65536 個有ります。それぞれの記憶場所に番号が振ってあります。その番号を番地と言い、#0000~#FFFF 番地まで有ります。これも、16 ビットです。

1つの番地には 16 ビットのデータを記憶することが出来ます。

³ ただし、CALL や RET の場合はこの限りではない。今後の学習範囲。

COMET II の説明には、普通含みませんが、コンピューターを考える場合、次のものが必ずあります。

アドレスバス	CPU が主記憶装置にデータを書く場合 (WR=1)、各場所を示すための線です。あるいは、CPU が主記憶装置からデータを読み込む場合 (RD=1)、そのアドレスを指定するための線です。アドレスは 16 ビットなので、当然その線の数も 16 本になります。
データバス	CPU とメモリーとの間で、データを受け渡しするための線です。COMET II のデータは全ていつも 16 ビットなので、16 本の線があります。
WR 線	CPU が主記憶装置にデータを書き込む場合、その線を 1 にします。
RD 線	CPU が主記憶装置からデータを引き出す場合、その線を 1 にします。

3.3 COMET II の動作

次に COMET II の動作の仕組みを考えてみましょう。COMET II に限らず、どんなコンピューターでも基本的には同じです。

図 2 の状態に主記憶装置がなっていたとしよう。そして、プログラムレジスタの値が #0020 となっていたとしよう。主記憶装置のアドレス #0020 以降は、以下の CASL II のプログラムをアセンブルして機械語に直したものです。

```

PGM      START
          LD      GR0,A
          ADDA    GR0,B
          ST      GR0,C
          RET
A         DC      1
B         DC      1
C         DC      0
          END

```

それでは、動作を 1 ずつ手順を追って説明します。

- ① プログラムレジスタの中身が #0020 なので、アドレスバスが (0000 0000 0010 0000) となる。そして、RD=1 とする。すると、主記憶装置が、データバスを (0001 0000 0000 0000) とします。
- ② 受け取ったデータ #1000 を CPU が解析する。解析の結果、その命令は 2 語と分かるので、RD 線とアドレスバス、データバスを使って、CPU からもう 1 語分 #0027 を読み出します。
- ③ 2 語読み出したので、プログラムレジスタを +2 加算する。即ち、PR=#0022 となる。
- ④ 読み出した命令 (#1000 と #0027) から、#0027 番地のデータを読み出して、汎用レジスタ GR0 にその値を入れる。GR0=#0001 となる。そして、フラグレジスタをセット (OF=0, SF=0, ZF=0) する。これで最初の命令が完了。
- ⑤ 次に、プログラムレジスタ PR=#0022 に従い、その番地の命令 #2000 を読み出す。

- ⑥ この命令を CPU が解析して、2 語と分かる。そして、#0023 番地のデータ #0028 を主記憶装置から読み出す。
- ⑦ 2 語読み出したので、プログラムレジスタを+2 加算する。即ち、PR=#0024 となる。
- ⑧ 読み出した命令(#2000 と#0028)から、#0028 番地のデータを読み出して、汎用レジスタ GR0 との和を計算して、結果を汎用レジスタに戻す。GR0=#0002 となる。そして、フラグレジスタをセット (OF=0, SF=0, ZF=0) する。これで 2 番目の命令が完了。
- ⑨ プログラムレジスタ PR=#0024 に従い、その番地の命令#1100 を読み出す。
- ⑩ この命令を CPU が解析して、2 語と分かる。そして、#0025 番地のデータ #0029 を主記憶装置から読み出す。
- ⑪ 2 語読み出したので、プログラムレジスタを+2 加算する。即ち、PR=#0026 となる。
- ⑫ 読み出した命令(#1100 と#0029)から、汎用レジスタ GR0 の内容を#0029 番地のデータに書き込む。これで 3 番目の命令は終了。
- ⑬ プログラムレジスタ PR=#0026 に従い、その番地の命令#8100 を読み出す。
- ⑭ この命令を CPU が解析して、RET 命令と分かる。スタックポインタ (PR) が示すアドレスの値をプログラムレジスタ PR にセットする (この辺は後の学習範囲)。これでこの命令は終わり。

3.3 データとは

主記憶装置に格納されているデータは、16 ビットのただの数字です。いわば、16 個の 0 と 1 の集まりにすぎません。16 進数で書くと、4 桁の数字にすぎません。その 4 桁の 16 進数の数字が命令になったり、数値データであったり、文字の場合もあります。それはどのように区別しているのでしょうか?。実は、全く区別していません。ただいえる事は、プログラムレジスタ PR が示すアドレスのデータは命令と解釈されるということだけです。後は、その命令にしたがい、主記憶装置のデータが命令になったり、数値になったり、文字になったりします。

命令と処理すべきデータ (数値や文字) が同じところに、区別無く格納されています。これをノイマン型コンピュータをいいます。

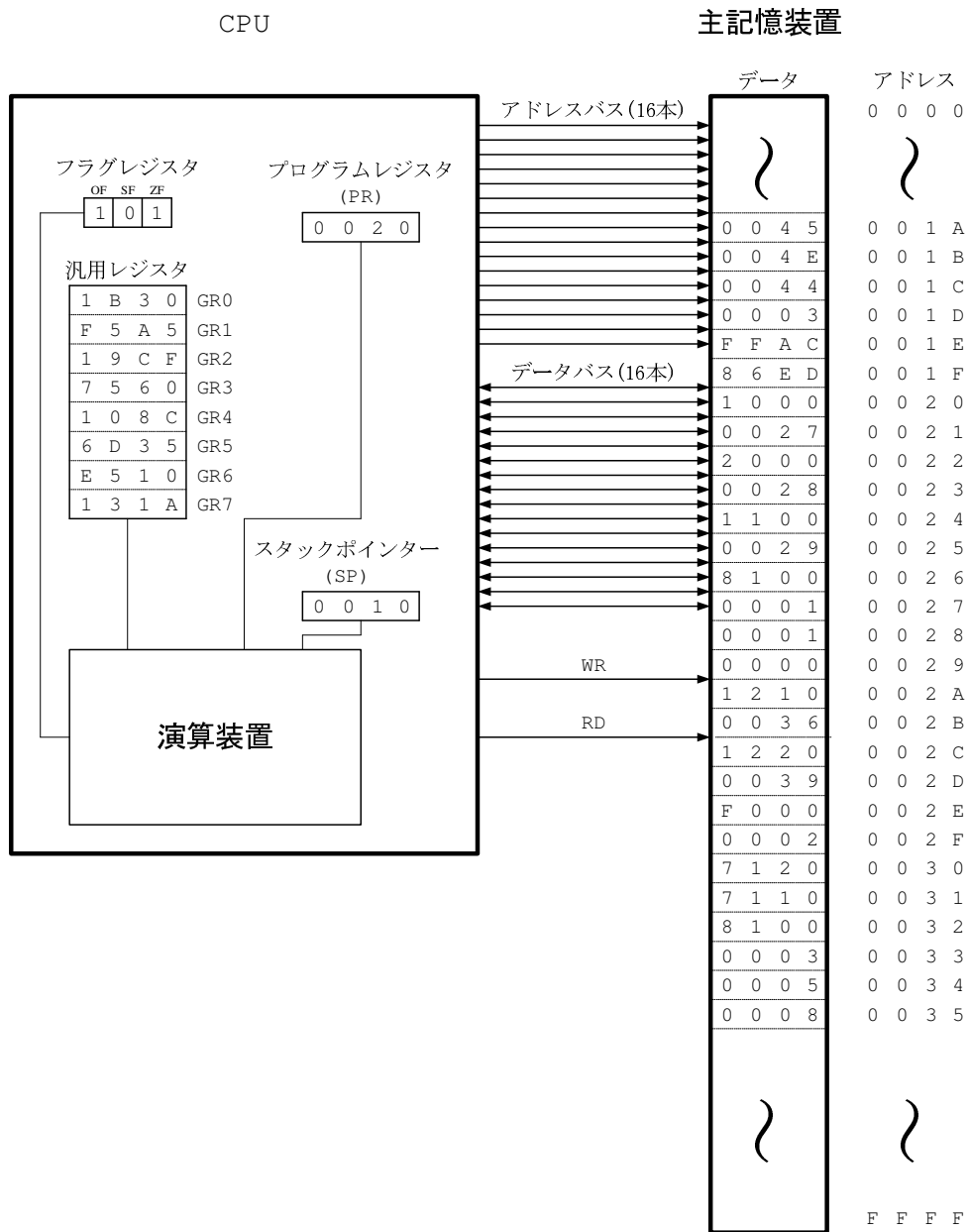


図2 COMET II の CPU と主記憶装置